



إدارة البحوث

الجمعية
العلمية
بكلية
التربية

كوبول

محمد بسام زهير الخياط
عضو هيئة التدريس بالمعهد

أحمد جنيد جنيد
عضو هيئة التدريس بالمعهد

المحتويات

الصفحة

١١

تقديم

١٥

القسم الأول : أساسيات لغة كوبول

١٧

الفصل ١ - : مقدمة

١٧

١ - ١ لمحة عامة

١٩

٢ - ١ الحاسب الآلي وبرنامج كوبول

٢٦

٣ - ١ الخطوات المنطقية لكتابة برنامج كوبول

٣١

٤ - ١ العناصر المكونة للغة كوبول

٣٧

٥ - ١ صفحة ترميز برنامج كوبول

٤١

٦ - ١ تمرينات

٤٣

الفصل ٢ - : التعليمات الأولية في لغة كوبول (قسم الاجراءات)

٤٣

١ - ٢ لمحة عامة

٤٣

٢ - ٢ قراءة صيغ لغة كوبول

٤٥

٣ - ٢ التعليمات الأولية للمعالجة ونقل التسلسل :

٤٥

OPEN

٢ - ٣ - ١ تعليمية فتح الملف

٤٦

READ

٢ - ٣ - ٢ تعليمية قراءة البيانات

٤٧

MOVE

٢ - ٣ - ٣ تعليمية نقل البيانات

٥٠

WRITE

٢ - ٣ - ٤ تعليمية كتابة السجل

٥١

CLOSE

٢ - ٣ - ٥ تعليمية غلق الملف

الصفحة

| | | | |
|----|----------|----------------------|-------|
| ٥٢ | GO TO | تعليلة نقل التسلسل | ٦-٣-٢ |
| ٥٢ | STOP | تعليلة التوقف | ٧-٣-٢ |
| ٥٣ | PERFORM | تعليلة نفذ (أنجز) | ٨-٣-٢ |
| ٦٢ | IF | تعليلة المقارنة | ٩-٣-٢ |
| ٦٤ | | التعليمات الحسائية : | ٤-٢ |
| ٦٥ | ADD | تعليلة الجمع | - |
| ٦٧ | SUBTRACT | تعليلة الطرح | - |
| ٦٩ | MULTIPLY | تعليلة الضرب | - |
| ٧٠ | DIVIDE | تعليلة التقسيم | - |
| ٧٣ | COMPUTE | تعليلة احسب | - |
| ٧٥ | | تمرينات | ٥-٢ |

الفصل - ٣ - الاقسام الأخرى في برنامج كويول

| | | | |
|----|-------------------------|------------------------|-------|
| ٧٩ | | لمحة عامة | ١-٣ |
| ٧٩ | IDENTIFICATION DIVISION | قسم التعريف | ٢-٣ |
| ٨١ | ENVIRONMENT DIVISION | قسم البيئة (المحيط) | ٣-٣ |
| ٨٣ | DATA DIVISION | قسم البيانات | ٤-٣ |
| ٨٣ | PICTURE | عبارة | ١-٤-٣ |
| ٨٤ | LEVEL NUMBERS | أرقام المستويات | ٢-٤-٣ |
| ٨٧ | FILE SECTION | فصل توصيف الملفات | ٣-٤-٣ |
| ٨٨ | WORKING-STORAGE SECTION | فصل التخزين الداخلي | ٤-٤-٣ |
| ٨٩ | VALUE CLAUSE | عبارة تحديد القيمة | ٥-٤-٣ |
| ٩١ | | الفاصلة العشرية | ٦-٤-٣ |
| ٩٢ | | إظهار البيانات الرقمية | ٧-٤-٣ |

الصفحة

| | |
|-----|-------------------------|
| ٩٥ | ٥-٣ برنامج كوبول متكامل |
| ١٠١ | ٦-٣ اسئلة وتمرينات |

القسم الثاني : لغة كوبرول المتقدمة

١٠٩

الفصل - ٤ - التعليمات المتقدمة في لغة كوبرول (قسم الاجراءات)

١١١

١١١ ١-٤ لحة عامة

١١١ ٢-٤ تعليمة المقارنة IF

١١٢ ١-٢-٤ اختبار نوع البيانات

١١٣ ٢-٢-٤ اختبار الإشارة الجبرية

١١٤ ٣-٢-٤ اختبار اسم الشرط

١١٦ COMPOUND IF ٤-٢-٤ المقارنة المركبة

١٢٣ NESTED IF ٥-٢-٤ المقارنة المتداخلة

١٢٦ CORRESPONDING ٣-٤ استخدام عبارة التطابق

١٣١ READ ... INTO ٤-٤ تعليمة القراءة في

١٣٢ WRITE ... FROM ٥-٤ تعليمة الكتابة من

١٣٣ ٦-٤ التقريب وكشف الخطأ في التعليمات الحساية

١٣٦ GO TO ... DEPENDING ON ٧-٤ تعليمات نقل التسلسل المحسوبة

١٣٨ DISPLAY ٨-٤ تعليمة عرض البيانات

١٤٢ ACCEPT ٩-٤ تعليمة قبول البيانات

١٤٧ INSPECT ١٠-٤ تعليمة فحص البيانات

١٥١ EXAMINE ١-١٠-٤ تعليمة فحص البيانات

١٥٢ ١١-٤ اسئلة وتمرينات

الفصل - ٥ - أقسام برنامج كوبرول الأخرى والعبارات المتقدمة

١٦١

١٦١ ١-٥ لحة عامة

الصفحة

| | | |
|-----|--------------|---------------------------------------|
| ١٦١ | | ٢-٥ رموز الإظهار (الاخراج) |
| ١٦٢ | SUPPRESSION | ١-٢-٥ الطمس البسيط |
| ١٦٣ | | ٢-٢-٥ الطمس مع الاستبدال |
| ١٦٤ | | ٣-٢-٥ الاستبدال العائم |
| ١٦٦ | INSERTION | ٤-٢-٥ الاقحام الثابت |
| ١٦٨ | | ٥-٢-٥ الفاصلة العشرية |
| ١٦٩ | | ٣-٥ عبارة إعادة التوصيف |
| ١٧٠ | | ٤-٥ توصيف الإشارة الجبرية |
| ١٧١ | SIGN | ١-٤-٥ عبارة الإشارة |
| ١٧٣ | USAGE | ٥-٥ عبارة صيغة التوصيف |
| ١٧٤ | DISPLAY | ١-٥-٥ صيغة |
| ١٧٥ | COMP-3 | ٢-٥-٥ صيغة |
| ١٧٧ | COMP | ٣-٥-٥ صيغة |
| ١٧٨ | RENAMES | ٦-٥ عبارة إعادة التسمية |
| ١٧٩ | COPY | ٧-٥ تعلية النسخ |
| ١٨١ | SEGMENTATION | ٨-٥ تجزئة قسم الاجراءات |
| ١٨٣ | | ٩-٥ البرامج الفرعية والبرامج الرئيسية |
| ١٩٣ | | ١٠-٥ اسئلة وتقرينات |

| | | |
|-----|--------|-----------------------------------|
| ٢٠١ | | الفصل - ٦ - الجداول في لغة كوبيول |
| ٢٠١ | | ١-٦ لمحة عامة |
| ٢٠٢ | OCCURS | ٢-٦ تعريف الجدول وعبارة التكرار |
| ٢٠٤ | | ٣-٦ تعريف الجدول ذو أكثر من بُعد |
| ٢٠٧ | | ٤-٦ تداول الجداول |

الصفحة

| | | | |
|-----|---------------------|-------------------------|-------|
| ٢٠٧ | SUBSCRIBING | استخدام المؤشر | ١-٤-٦ |
| ٢٠٩ | INDEXING | استخدام الدليل | ٢-٤-٦ |
| ٢١١ | USAGE IS INDEX | عبارة استخدام الدليل | ٣-٤-٦ |
| ٢١٤ | SET | تعلية تحديد قيمة الدليل | ٥-٦ |
| ٢١٦ | PERFORM ... VARYING | تعلية أنجز المتبدلة | ٦-٦ |
| ٢٢٠ | SEARCH | تعلية البحث في الجدول | ٧-٦ |
| ٢٢٢ | SEQUENTIAL | البحث التتابعي | ١-٧-٦ |
| ٢٢٤ | BINARY | البحث الثنائي | ٢-٧-٦ |
| ٢٢٦ | | امثلة وقرينات | ٨-٦ |

الفصل - ٧ - الفرز والدمج في لغة كوبرول

| | | |
|-----|---------------------|-----|
| ٢٣٣ | لمحة عامة | ١-٧ |
| ٢٣٣ | الفرز في لغة كوبرول | ٢-٧ |
| ٢٣٤ | دمج الملفات | ٣-٧ |
| ٢٤٣ | امثلة وقرينات | ٤-٧ |

الفصل - ٨ - الملفات ومعالجتها

| | | |
|-----|---|-------|
| ٢٥١ | لمحة عامة | ١-٨ |
| ٢٥١ | أنواع الملفات | ٢-٨ |
| ٢٥٥ | بنية الملفات | ٣-٨ |
| ٢٥٨ | تنظيم الملفات | ٤-٨ |
| ٢٥٩ | تنظيم الملفات على أوساط تخزين متتابعة | ١-٤-٨ |
| ٢٦٢ | تنظيم الملفات على أوساط تخزين مباشرة - المعالجة | ٢-٤-٨ |

الصفحة

| | |
|-----|--|
| ٢٧٢ | ٥-٨ أساليب معالجة الملفات |
| ٢٧٣ | ١-٥-٨ معالجة الملفات المخزنة على أوساط تخزين متتابعة |
| ٢٨١ | ٢-٥-٨ معالجة الملفات المخزنة على أوساط تخزين مباشرة - المعالجة |
| ٢٨٢ | ١-٢-٥-٨ المعالجة المتسلسلة |
| ٢٩٣ | ٢-٢-٥-٨ المعالجة العشوائية (المباشرة) |
| ٣١٢ | ٣-٢-٥-٨ DYNAMIC المعالجة الحركية |
| ٣١٥ | ٦-٨ أسئلة وتعارين |

| | |
|-----|---|
| ٣١٩ | الفصل - ٩ - البرمجة الهيكلية بلغة كوبرول |
| ٣١٩ | ١-٩ لمحة عامة |
| ٣٢٠ | ٢-٩ نظرية البرمجة الهيكلية STRUCTURED |
| ٣٢٣ | ٣-٩ أشكال البرمجة الهيكلية بلغة كوبرول |
| ٣٢٥ | ٤-٩ إرشادات في ترميز برنامج كوبرول على ضوء البرمجة الهيكلية |
| ٣٢٨ | ٥-٩ مثال عملي |
| ٣٣٥ | ٦-٩ أسئلة وتقرينات |

| | |
|-----|--------------------------------------|
| ٣٤١ | الفصل - ١٠ - تدقيق البرنامج واختباره |
| ٣٤١ | ١-١٠ لمحة عامة |
| ٣٤١ | ٢-١٠ الأخطاء الشكلية |
| ٣٤٢ | ٣-١٠ الأخطاء المنطقية |
| ٣٤٣ | ٤-١٠ أخطاء شائعة |
| ٣٤٤ | ٥-١٠ كشف الأخطاء وتصحيحها |
| ٣٤٥ | ٦-١٠ أسئلة وتقرينات |

القسم الثالث : استخدام الحاسب في

كتابة وترجمة وتنفيذ برنامج كوبرول

٣٦٣

الفصل - ١١ - استخدام منقح النصوص TEXT EDITOR

٣٦٥

٣٦٥

٣٦٦

٣٦٦

٣٧٤

٣٧٧

٣٧٧

٣٨٦

٤٢٤

٤٢٤

٤٢٤

٤٢٦

١-١١ لوحة عامة

٢-١٠ استخدام نظام الحاسب I B M 3033

١-٢-١١ العمليات الأولية

٢-٢-١١ تشغيل النماذج من نوع 3278

٣-١١ استخدام منقح النصوص بواسطة الشاشة 3278

١-٣-١١ كتابة برنامج جديد

٢-٣-١١ تنقيح برنامج / ملف موجود سابقاً

٤-١١ استخدام نظام الحاسب TI990

١-٤-١١ استخدام منقح النصوص في الحاسب TI990

١-٤-١١ كتابة برنامج جديد

٢-٤-١١ تنقيح برنامج / ملف موجود سابقاً

الفصل - ١٢ - ترجمة البرنامج وتنفيذه

٤٣٣

٤٣٣

٤٣٣

٤٤٦

١-١٢ لوحة عامة

٢-١٢ ترجمة وتنفيذ البرنامج باستخدام الحاسب

IBM 3033 / نظام التشغيل MVS

٣-١٢ ترجمة وتنفيذ البرنامج باستخدام الحاسب

TI990 / نظام التشغيل DXIO

الصفحة

٤٥١

القسم الرابع : تطبيقات عملية

٤٥٣ الفصل ١٣ - تطبيقات عملية للعمليات الأساسية في برمجة كوبول

٤٥٣ ١-١٣ لمحة عامة

٤٥٣ ٢-١٣ برنامج طباعة تقرير

٤٥٤ - نص التمرين

٤٥٦ - نص البرنامج ونتيجته

٤٦١ ٣-١٣ برنامج معالجة بيانات / الأسلوب المباشر

٤٦٣ - نص التمرين

٤٦٦ - نص البرنامج

٤٧٢ ٤-١٣ برنامج معالجة بيانات وطباعة تقرير

٤٧٢ - نص التمرين

٤٧٥ - نص البرنامج ونتيجته

٤٨٥ ٥-١٣ تمرينات عامة

٤٩٥ الفصل ١٤ - المراجع

٤٩٨ الفصل ١٥ - ملحقات :

٤٩٩ ١-١٥ جدول رموز الاخراج

٥٠٠ ٢-١٥ كلمات لغة كوبول المحجوزة

٥٠٤ ٣-١٥ الصيغ العامة في لغة كوبول

٥١٧ ٤-١٥ معجم لأهم المصطلحات المستخدمة في لغة كوبول

٥٤١ ٥-١٥ حلول مقترحة لبعض التمارين

٥٩٣ أشكال مخارج ونتائج بعض التمارين

تشكل التطبيقات الإدارية والتجارية على الحاسبات الالكترونية نسبة كبيرة بالمقارنة مع التطبيقات الأخرى ، ولذلك كان للغة البرمجة كوبرول دور واسع وكبير في كتابة وتطوير البرامج والأنظمة ذات الطبيعة الإدارية والتجارية لأنها مصممة ومطورة لتفي بهذا الغرض بشكل رئيسي .

وانطلاقاً من هذا الاستخدام الواسع لهذه اللغة العالية من لغات البرمجة وتمشياً مع التطوير السريع والنهضة العلمية في المملكة فقد قمنا باعداد هذا الكتاب ليكون مساعداً لقراء اللغة العربية في فهم لغة البرمجة هذه والتعرف على كيفية استخدامها .

ولا تنحصر الاستفادة من هذا المؤلف بالدارسين في دبلوم الحاسب الآلي في المعهد بل يمكن أن يكون مرجعاً لهذه اللغة والتدرب عليها في مختلف المستويات .

وقد اعتمدنا عند وضع هذا المؤلف في جميع أقسامه وبفصوله على لغة كوبرول المطورة عام ١٩٧٤ والتي وافقت عليها هيئة ANSI للغات . علاوة على الأخذ بعين الاعتبار المواضيع التي طورت بعد حلول الحاسبات الصغيرة Mini Computer والتي تعتمد على الأسلوب المباشر في التشغيل ، مما يجعل المؤلف مفيداً في التدريب على كلا النوعين من الحاسبات الصغيرة والكبيرة وطريقتي التشغيل المباشرة وغير المباشرة .

وقد روعي عند وضع هذا المؤلف الغرض التدريبي والتعليمي منه وذلك بالجوء الى تسلسل خاص للمواضيع المطروقة . والذي قد يختلف مع المؤلفات الماثلة ، بحيث يمكن للدارس استخدام الحاسب والتدرب عليه وفق هذه اللغة بأقصر مدة ممكنة . ولذلك فقد تم التركيز على قسم الاجراءات في برنامج كوبول واعطاؤه الأولية في الشرح والبحث لأنه جوهر البرنامج وأساسه المنطقي .

وقد تم تقسيم هذا المؤلف إلى أربعة أقسام رئيسية :

القسم الأول : ويتكون من ثلاثة فصول تبحث في أساسيات لغة كوبول وأقسام البرنامج الأساسية وكيفية كتابة برنامج بسيط بهذه اللغة .

القسم الثاني : ويتكون من ستة فصول تبحث في التعليمات والعبارات المتقدمة في لغة كوبول مما يعطي المبرمج المبتدئ المعلومات اللازمة لتحسين وتطوير برامجه البسيطة .

القسم الثالث : ويتكون من فصلين ويحتوي على المواضيع الخاصة باستخدام الحاسب في كتابة وترجمة وتنفيذ برنامج كوبول .

القسم الرابع : ويتكون من فصل واحد ويحتوي على تطبيقات عملية شاملة لمختلف المواضيع المهمة والأساسية في هذه اللغة مثل انشاء الملفات وتعديلها وطباعتها .

كما اشتمل المؤلف على ملحقات خاصة بالموضوعات الضرورية لتساعد المبرمج في الرجوع اليها عند كتابة برامج كوبول .

ونتوجه بشكرنا وتقديرنا الى إدارة المعهد للتشجيع والمساعدة الذين لاقيناهما عند إعداد هذا المؤلف وإبرازه الى حيز الوجود .

كما أننا نشكر قسم الطباعة والنشر وكافة العاملين فيه على الجهد الكبير اللازم
لاظهار هذا المؤلف بالشكل اللائق والمطلوب.

ونأمل أن نكون قد وفقنا في وضع هذا المؤلف ليكون عوناً للدارسين والمهتمين
والطلاب في فهم واستخدام لغة كوبرول وليكون لبنة من اللبنة الأولى في بناء صرح
المكتبة العربية المعاصرة.

والله من وراء القصد

٢٢ رمضان ١٤٠٣هـ

٣ تموز ١٩٨٣م

المؤلفان

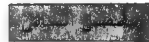


أساسيات لغة كوبيول

مقدمة



التعليمات الأولية في لغة الكوبيول
(قسم الإجراءات)



الأقسام الأخرى في برنامج كوبيول



الفصل الأول

مقدمة

١ - ١ لمحة عامة *

تعتبر لغة الكوبول (COBOL) المختصرة من الكلمات الانكليزية التالية Common Business Oriented Language (وتعني اللغة الموجهة لإدارة الأعمال العامة) من أوسع لغات البرمجة العالية انتشاراً. وقد صنفت مع اللغات العالية بسبب سهولة قراءتها وقربها الى اللغة الانكليزية العادية، ويمكن كتابة البرامج بها بعيداً عن الرموز والاصطلاحات الغامضة.

وقد بدأ العمل في تطوير لغة الكوبول تاريخياً منذ عام ١٩٥٩ ميلادي (مايو) في الولايات المتحدة من قبل المستفيدين في حقل إدارة الأعمال من الحاسب الآلي. وصدرت النسخة الأولى (Version) في نفس العام في شهر ديسمبر ثم تلتها النسخة المنقحة عام ١٩٦١ ميلادي وسميت COBOL - 61 والتي تعتبر حجر الأساس للتحسينات والتطوير الذي طرأ على النسخ التالية. وفي عام ١٩٦٨ ميلادي تمت الموافقة على النسخة المعروفة باسم ANSI COBOL من قبل هيئة تسمى American National Standard Institute.

وبعد ذلك استمرت التحسينات والإضافات على لغة الكوبول وكذلك حذفت الأشياء الغامضة وغير الواضحة بحيث أقرت هيئة ANSI النسخة الجديدة والأخيرة عام ١٩٧٤ وسميت ANSI COBOL 74.

(*) Structured COBOL, Philipppakis and Kazmier, MacGraw Hill, 1981 P. 17, 18.

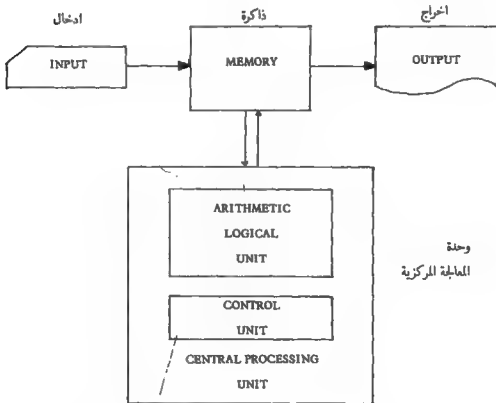
وتعتبر الميزة الأساسية والأولى للغة كوبول أنها لغة موثقة ذاتياً أي Self-documented لأن المرونة المتوفرة في اختيار أسماء الحقول والمتغيرات كبيرة . كذلك لكونها قريبة جداً من اللغة الانكليزية . وهذه الميزة تسمح لمراجع البرنامج بفهمه ومعرفة وظائفه بسرعة وكذلك مهمة جداً لأجل صيانة وتطوير البرنامج في المستقبل مما يجنب المبرمج إعادة كتابة نفس البرنامج عند عدم وضوحه ومعرفة وظائفه من أجل عملية التطوير والتحسين على البرنامج التي تعتبر أساسية في عملية البرمجة . ولكن الى جانب إمكانية كتابة برنامج واضح وسهل ومقروء بلغة كوبول يمكن كتابة برامج غامضة عند استخدام أسماء غير واضحة ورموز بدلا من الأسماء المعبرة عن المتغيرات والحقول المستخدمة في البرنامج .

وعلى الرغم من كون لغة كوبول مستخدمة منذ زمن بعيد نسبياً إلا أن استخدامها قد زاد في فترة السبعينات (١٩٧٠م وما بعد) . وانتشارها الواسع بعد ذلك التاريخ أدى الى ظهور نسخ (Versions) محسنة من هذه اللغة تتمشى مع ظهور الحاسبات الصغيرة Mini Computer وحلول الشاشة المتوفرة «VDU» لاستقبال وإظهار البيانات . ولكن تبقى الاقسام الأساسية في لغة كوبول ثابتة ، وهذه التحسينات تعتبر إضافات ثانوية الى حين اعتمادها من قبل الهيئة المسماة ANSI .

وتجدر الإشارة أخيراً إلى أن عملية كتابة البرامج بلغة كوبول كانت غير مستحبة في السابق لأنها تحتاج الى وقت أطول اثناء التنفيذ على الحاسب ولكن بعد تطوير المترجمات COBOL COMPILERS أصبحت سرعة تنفيذ البرامج المكتوبة بهذه اللغة قريبة الى نفس السرعة التي تحتاجها البرامج المكتوبة بلغة ذات مستوى أدنى Low-Level . ولذلك تعتبر لغة كوبول اليوم من اللغات الأساسية في معالجة البيانات الإدارية والأكثر شيوعاً في الاستخدام وسوف تبقى كذلك لحين ظهور لغة برمجة لها نفس الصفات وبفعالية أكبر .

١-٢ الحاسب الآلي وبرنامج كوبول :

من المعروف أن الحاسب الآلي عبارة عن مجموعة من الأجهزة الالكترونية التي تقبل المعلومات وتعالجها وترصد نتائجها وفق خطة محددة تسمى البرنامج. ويتألف الحاسب الآلي من مجموعة من الأجهزة أو الوحدات تسمى وحدات إدخال أو وحدات إخراج ومن وحدة المعالجة المركزية كما هو مبين في الشكل التخطيطي التالي :



والبرنامج عبارة عن مجموعة من التعليمات المتسلسلة تسلسلاً منطقياً لأداء غرض معين، فهو يقرأ البيانات من وحدات إدخال مثلاً ويعالج هذه البيانات ثم يظهر نتائجها على وحدات إخراج وهكذا ..

ويمكن للبرنامج أن يصاغ بلغات حاسب مختلفة، هذه اللغات تنقسم الى قسمين :

- لغات دنيا LOW-LEVEL LANGUAGES مثل اسمبلر ASSEMBLER .
- لغات عليا HIGH-LEVEL LANGUAGES مثل كوبول، فورتران، بيزك....

إن لكل لغة مزايا خاصة، و يتحكم في اختيار لغة البرنامج عدة اعتبارات أهمها طبيعة النظام أو البرنامج. وتعتبر لغة كوبول من لغات البرمجة العليا، والبرامج المكتوبة بهذه اللغة تعتبر بصيغة المصدر أي SOURCE وتحول الى لغة دنيا خاصة بالحاسب عبر إمرارها على برنامج مترجم يسمى - المترجم COMPILER - الذي يحولها إلى صيغة التنفيذ OBJECT*.

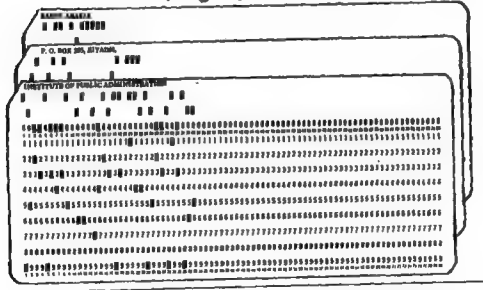
ولإعطاء فكرة سريعة وواضحة من برنامج كوبول نأخذ المثال التالي :

لدينا ملف من البطاقات المثقبة يحتوي على اسم معهد الادارة العامة وعنوانه :

INSTITUTE OF PUBLIC ADMINISTRATION

P.O. BOX 205, RIYADH, SAUDI ARABIA

المطلوب قراءة هذه البطاقات وطباعة محتواها على الطابعة.



(*) تمثل لغة الآلة القابلة للتنفيذ بعد الترجمة

ولا بد قبل وضع أي برنامج من رسم مخططه الانسيابي أو المنطقي flowchart

وقبل البدء في رسم المخطط الانسيابي للمسألة، نتعرف على الرموز المستخدمة في رسم المخطط الانسيابي وهي :



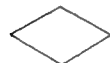
START / END رمز البداية أو النهاية



PROCESS رمز المعالجة



READ / WRITE رمز القراءة أو الكتابة



DECISION رمز اتخاذ قرار



CONNECTOR صلة وصل



DIRECTION OF FLOW اتجاه التدفق أو السير

الرموز المستخدمة حسب

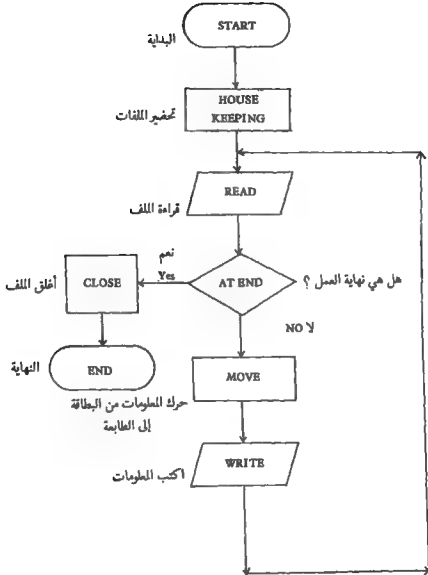
AMERICAN NATIONAL STANDARDS INSTITUTE FLOWCHART SYMBOLS

الشكل رقم (١-١)

ومن استخدام هذه الرموز يمكن وضع أي مخطط انسيابي لبرنامج حاسب آلي فيكون المخطط المطلوب هو التالي والذي يوضح سير البرنامج منطقياً .

(٩) المحجم العربي الموحد لمصطلحات الحاسبات الالكترونية - المنظمة العربية للعلوم الإدارية بجامعة الدول العربية - عمان ١٩٨١، ص ١٢٨ .

المخطط الانسيابي للمسألة :



الشكل رقم (١-٢)

وفي الصفحة التالية ترميز لهذا المخطط الانسيابي وفق قواعد لغة كوبرول .

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | AA | AB | AC | AD | AE | AF | AG | AH | AI | AJ | AK | AL | AM | AN | AO | AP | AQ | AR | AS | AT | AU | AV | AW | AX | AY | AZ | BA | BB | BC | BD | BE | BF | BG | BH | BI | BJ | BK | BL | BM | BN | BO | BP | BQ | BR | BS | BT | BU | BV | BW | BX | BY | BZ | CA | CB | CC | CD | CE | CF | CG | CH | CI | CJ | CK | CL | CM | CN | CO | CP | CQ | CR | CS | CT | CU | CV | CW | CX | CY | CZ | DA | DB | DC | DD | DE | DF | DG | DH | DI | DJ | DK | DL | DM | DN | DO | DP | DQ | DR | DS | DT | DU | DV | DW | DX | DY | DZ | EA | EB | EC | ED | EE | EF | EG | EH | EI | EJ | EK | EL | EM | EN | EO | EP | EQ | ER | ES | ET | EU | EV | EW | EX | EY | EZ | FA | FB | FC | FD | FE | FF | FG | FH | FI | FJ | FK | FL | FM | FN | FO | FP | FQ | FR | FS | FT | FU | FV | FW | FX | FY | FZ | GA | GB | GC | GD | GE | GF | GG | GH | GI | GJ | GK | GL | GM | GN | GO | GP | GQ | GR | GS | GT | GU | GV | GW | GX | GY | GZ | HA | HB | HC | HD | HE | HF | HG | HH | HI | HJ | HK | HL | HM | HN | HO | HP | HQ | HR | HS | HT | HU | HV | HW | HX | HY | HZ | IA | IB | IC | ID | IE | IF | IG | IH | II | IJ | IK | IL | IM | IN | IO | IP | IQ | IR | IS | IT | IU | IV | IW | IX | IY | IZ | JA | JB | JC | JD | JE | JF | JG | JH | JI | IJ | JK | KL | KM | KN | KO | KP | KQ | KR | KS | KT | KU | KV | KW | KX | KY | KZ | LA | LB | LC | LD | LE | LF | LG | LH | LI | LJ | LK | LM | LN | LO | LP | LQ | LR | LS | LT | LU | LV | LW | LX | LY | LZ | MA | MB | MC | MD | ME | MF | MG | MH | MI | MJ | MK | ML | MM | MN | MO | MP | MQ | MR | MS | MT | MU | MV | MW | MX | MY | MZ | NA | NB | NC | ND | NE | NF | NG | NH | NI | NJ | NK | NL | NM | NN | NO | NP | NQ | NR | NS | NT | NU | NV | NW | NX | NY | NZ | OA | OB | OC | OD | OE | OF | OG | OH | OI | OJ | OK | OL | OM | ON | OO | OP | OQ | OR | OS | OT | OU | OV | OW | OX | OY | OZ | PA | PB | PC | PD | PE | PF | PG | PH | PI | PJ | PK | PL | PM | PN | PO | PP | PQ | PR | PS | PT | PU | PV | PW | PX | PY | PZ | QA | QB | QC | QD | QE | QF | QG | QH | QI | QJ | QK | QL | QM | QN | QO | QP | QQ | QR | QS | QT | QU | QV | QW | QX | QY | QZ | RA | RB | RC | RD | RE | RF | RG | RH | RI | RJ | RK | RL | RM | RN | RO | RP | RQ | RR | RS | RT | RU | RV | RW | RX | RY | RZ | SA | SB | SC | SD | SE | SF | SG | SH | SI | SJ | SK | SL | SM | SN | SO | SP | SQ | SR | SS | ST | SU | SV | SW | SX | SY | SZ | TA | TB | TC | TD | TE | TF | TG | TH | TI | TJ | TK | TL | TM | TN | TO | TP | TQ | TR | TS | TT | TU | TV | TW | TX | TY | TZ | UA | UB | UC | UD | UE | UF | UG | UH | UI | UJ | UK | UL | UM | UN | UO | UP | UQ | UR | US | UT | UU | UV | UW | UX | UY | UZ | VA | VB | VC | VD | VE | VF | VG | VH | VI | VJ | VK | VL | VM | VN | VO | VP | VQ | VR | VS | VT | VU | VV | VW | VX | VY | VZ | WA | WB | WC | WD | WE | WF | WG | WH | WI | WJ | WK | WL | WM | WN | WO | WP | WQ | WR | WS | WT | WU | WV | WW | WX | WY | WZ | XA | XB | XC | XD | XE | XF | XG | XH | XI | XJ | XK | XL | XM | XN | XO | XP | XQ | XR | XS | XT | XU | XV | XW | XX | XY | XZ | YA | YB | YC | YD | YE | YF | YG | YH | YI | YJ | YK | YL | YM | YN | YO | YP | YQ | YR | YS | YT | YU | YV | YW | YX | YY | YZ | ZA | ZB | ZC | ZD | ZE | ZF | ZG | ZH | ZI | ZJ | ZK | ZL | ZM | ZN | ZO | ZP | ZQ | ZR | ZS | ZT | ZU | ZV | ZW | ZX | ZY | ZZ | AA | AB | AC | AD |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|

| A | | B | | C | | D | | E | | F | | G | | H | | I | | J | | K | | L | | M | | N | | O | | P | | Q | | R | | S | | T | | U | | V | | W | | X | | Y | | Z | | AA | | AB | | AC | | AD | | AE | | AF | | AG | | AH | | AI | | AJ | | AK | | AL | | AM | | AN | | AO | | AP | | AQ | | AR | | AS | | AT | | AU | | AV | | AW | | AX | | AY | | AZ | | BA | | BB | | BC | | BD | | BE | | BF | | BG | | BH | | BI | | BJ | | BK | | BL | | BM | | BN | | BO | | BP | | BQ | | BR | | BS | | BT | | BU | | BV | | BW | | BX | | BY | | BZ | | CA | | CB | | CC | | CD | | CE | | CF | | CG | | CH | | CI | | CJ | | CK | | CL | | CM | | CN | | CO | | CP | | CQ | | CR | | CS | | CT | | CU | | CV | | CW | | CX | | CY | | CZ | | DA | | DB | | DC | | DD | | DE | | DF | | DG | | DH | | DI | | DJ | | DK | | DL | | DM | | DN | | DO | | DP | | DQ | | DR | | DS | | DT | | DU | | DV | | DW | | DX | | DY | | DZ | | EA | | EB | | EC | | ED | | EE | | EF | | EG | | EH | | EI | | EJ | | EK | | EL | | EM | | EN | | EO | | EP | | EQ | | ER | | ES | | ET | | EU | | EV | | EW | | EX | | EY | | EZ | | FA | | FB | | FC | | FD | | FE | | FF | | FG | | FH | | FI | | FJ | | FK | | FL | | FM | | FN | | FO | | FP | | FQ | | FR | | FS | | FT | | FU | | FV | | FW | | FX | | FY | | FZ | | GA | | GB | | GC | | GD | | GE | | GF | | GG | | GH | | GI | | GJ | | GK | | GL | | GM | | GN | | GO | | GP | | GQ | | GR | | GS | | GT | | GU | | GV | | GW | | GX | | GY | | GZ | | HA | | HB | | HC | | HD | | HE | | HF | | HG | | HH | | HI | | HJ | | HK | | HL | | HM | | HN | | HO | | HP | | HQ | | HR | | HS | | HT | | HU | | HV | | HW | | HX | | HY | | HZ | | IA | | IB | | IC | | ID | | IE | | IF | | IG | | IH | | II | | IJ | | IK | | IL | | IM | | IN | | IO | | IP | | IQ | | IR | | IS | | IT | | IU | | IV | | IW | | IX | | IY | | IZ | | JA | | JB | | JC | | JD | | JE | | JF | | JG | | JH | | JI | | JJ | | JK | | JL | | JM | | JN | | JO | | JP | | JQ | | JR | | JS | | JT | | JU | | JV | | JW | | JX | | JY | | JZ | | KA | | KB | | KC | | KD | | KE | | KF | | KG | | KH | | KI | | KJ | | KK | | KL | | KM | | KN | | KO | | KP | | KQ | | KR | | KS | | KT | | KU | | KV | | KW | | KX | | KY | | KZ | | LA | | LB | | LC | | LD | | LE | | LF | | LG | | LH | | LI | | LJ | | LK | | LL | | LM | | LN | | LO | | LP | | LQ | | LR | | LS | | LT | | LU | | LV | | LW | | LX | | LY | | LZ | | MA | | MB | | MC | | MD | | ME | | MF | | MG | | MH | | MI | | MJ | | MK | | ML | | MM | | MN | | MO | | MP | | MQ | | MR | | MS | | MT | | MU | | MV | | MW | | MX | | MY | | MZ | | NA | | NB | | NC | | ND | | NE | | NF | | NG | | NH | | NI | | NJ | | NK | | NL | | NM | | NN | | NO | | NP | | NQ | | NR | | NS | | NT | | NU | | NV | | NW | | NX | | NY | | NZ | | OA | | OB | | OC | | OD | | OE | | OF | | OG | | OH | | OI | | OJ | | OK | | OL | | OM | | ON | | OO | | OP | | OQ | | OR | | OS | | OT | | OU | | OV | | OW | | OX | | OY | | OZ | | PA | | PB | | PC | | PD | | PE | | PF | | PG | | PH | | PI | | PJ | | PK | | PL | | PM | | PN | | PO | | PP | | PQ | | PR | | PS | | PT | | PU | | PV | | PW | | PX | | PY | | PZ | | QA | | QB | | QC | | QD | | QE | | QF | | QG | | QH | | QI | | QJ | | QK | | QL | | QM | | QN | | QO | | QP | | QQ | | QR | | QS | | QT | | QU | | QV | | QW | | QX | | QY | | QZ | | RA | | RB | | RC | | RD | | RE | | RF | | RG | | RH | | RI | | RJ | | RK | | RL | | RM | | RN | | RO | | RP | | RQ | | RR | | RS | | RT | | RU | | RV | | RW | | RX | | RY | | RZ | | SA | | SB | | SC | | SD | | SE | | SF | | SG | | SH | | SI | | SJ | | SK | | SL | | SM | | SN | | SO | | SP | | SQ | | SR | | SS | | ST | | SU | | SV | | SW | | SX | | SY | | SZ | | TA | | TB | | TC | | TD | | TE | | TF | | TG | | TH | | TI | | TJ | | TK | | TL | | TM | | TN | | TO | | TP | | TQ | | TR | | TS | | TT | | TU | | TV | | TW | | TX | | TY | | TZ | | UA | | UB | | UC | | UD | | UE | | UF | | UG | | UH | | UI | | UJ | | UK | | UL | | UM | | UN | | UO | | UP | | UQ | | UR | | US | | UT | | UU | | UV | | UW | | UX | | UY | | UZ | | VA | | VB | | VC | | VD | | VE | | VF | | VG | | VH | | VI | | VJ | | VK | | VL | | VM | | VN | | VO | | VP | | VQ | | VR | | VS | | VT | | VU | | VV | | VW | | VX | | VY | | VZ | | WA | | WB | | WC | | WD | | WE | | WF | | WG | | WH | | WI | | WJ | | WK | | WL | | WM | | WN | | WO | | WP | | WQ | | WR | | WS | | WT | | WU | | WV | | WW | | WX | | WY | | WZ | | XA | | XB | | XC | | XD | | XE | | XF | | XG | | XH | | XI | | XJ | | XK | | XL | | XM | | XN | | XO | | XP | | XQ | | XR | | XS | | XT | | XU | | XV | | XW | | XX | | XY | | XZ | | YA | | YB | | YC | | YD | | YE | | YF | | YG | | YH | | YI | | YJ | | YK | | YL | | YM | | YN | | YO | | YP | | YQ | | YR | | YS | | YT | | YU | | YV | | YW | | YX | | YY | | YZ | | ZA | | ZB | | ZC | | ZD | | ZE | | ZF | | ZG | | ZH | | ZI | | ZJ | | ZK | | ZL | | ZM | | ZN | | ZO | | ZP | | ZQ | | ZR | | ZS | | ZT | | ZU | | ZV | | ZW | | ZX | | ZY | | ZZ | |
|--------------------------|--|-------------|--|-------------------------|--|-------------------|--|-----------------------|--|------------------------|--|------------------|--|------------------|--|-----------------------|--|---------------|--|------------------------------------|--|-------------------------------------|--|----------------|--|---------------|--|---|--|--------------------------|--|------------------------|--|---|--|--------------------------|--|---------------------------|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|-----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|
| 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | 11 | | 12 | | 13 | | 14 | | 15 | | 16 | | 17 | | 18 | | 19 | | 20 | | 21 | | 22 | | 23 | | 24 | | 25 | | 26 | | 27 | | 28 | | 29 | | 30 | | 31 | | 32 | | 33 | | 34 | | 35 | | 36 | | 37 | | 38 | | 39 | | 40 | | 41 | | 42 | | 43 | | 44 | | 45 | | 46 | | 47 | | 48 | | 49 | | 50 | | 51 | | 52 | | 53 | | 54 | | 55 | | 56 | | 57 | | 58 | | 59 | | 60 | | 61 | | 62 | | 63 | | 64 | | 65 | | 66 | | 67 | | 68 | | 69 | | 70 | | 71 | | 72 | | 73 | | 74 | | 75 | | 76 | | 77 | | 78 | | 79 | | 80 | | 81 | | 82 | | 83 | | 84 | | 85 | | 86 | | 87 | | 88 | | 89 | | 90 | | 91 | | 92 | | 93 | | 94 | | 95 | | 96 | | 97 | | 98 | | 99 | | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IDENTIFICATION DIVISION. | | PROGRAM-ID. | | THIS PROGRAM PRINTS IPN | | NAME AND ADDRESS. | | ENVIRONMENT DIVISION. | | CONFIGURATION SECTION. | | SOURCE-COMPUTER. | | OBJECT-COMPUTER. | | INPUT-OUTPUT SECTION. | | FILE-CONTROL. | | SELECT CARD-F ASSIGN TO UT-S-CARD. | | SELECT PRINT-F ASSIGN TO UT-S-PRFL. | | DATA DIVISION. | | FILE SECTION. | | FD CARD-F RECORD CONTAINS 80 CHARACTERS | | LABEL RECORD IS OMITTED. | | 01 CARD-REC PIC X(80). | | FD PRINT-F RECORD CONTAINS 132 CHARACTERS | | LABEL RECORD IS OMITTED. | | 01 PRINT-LINE PIC X(132). | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

و يلاحظ من هذا المثال أن البرنامج يتألف من أربعة أقسام رئيسية وهي :

١ - قسم التعريف IDENTIFICATION DIVISION

و يستخدم هذا القسم لتعريف البرنامج باسم محدد وكذلك اسم كاتب البرنامج و يبدأ المبرمج كتابة هذه الفقرات من القسم A من العمود ٨ على صفحة الترميز الخاصة بلغة كوبول.

٢ - قسم البيئة (المحيط) ENVIRONMENT DIVISION

وهذا القسم يحتوي على توصيف الحاسب المستخدم وكذلك وحدات الإدخال والإخراج وذلك من أجل ربط الأسماء المذكورة في البرنامج بهذه الوحدات (الأسماء المنطقية) مع الأسماء الفعلية المعرفة في نظام التشغيل . و يقوم هذا القسم في البرنامج بدور الوسيط بين البرنامج نفسه ونظام التشغيل . لذلك لا بد من الرجوع الى المراجع المتوفرة عن الحاسب المستخدم أثناء كتابة هذا القسم .

قسم البيانات DATA DIVISION

يحتوي هذا القسم توصيف وتحديد البيانات المستخدمة في البرنامج ويمكن أن يشتمل على ما يلي :

- وصف بيانات الادخال والاخراج من / إلى البرنامج ضمن الفصل File Section وكلمة FD تعني File Description و يلاحظ أنها تبدأ من العمود ٨ - ١١ و بعد تعريف الملف يرد ذكر لوصف سجل من هذا الملف المستخدم برقم المستوى 01 .
- وصف للحقول أو الأماكن المحجوزة في الذاكرة لغرض الاستعمال الداخلي في البرنامج فصل WORKING-STORAGE ويحتوي في هذا المثال على الحقل
END - OF - JOB و يأخذ رقم المستوى 77 .

و يلاحظ أن أرقام المستوى وأسماء الفصول في هذا القسم تبدأ من العمود ٨ - ١١ في صفحة ترميز كوبول .

٤ - قسم الاجراءات PROCEDURE DIVISION

يحتوي هذا القسم على جوهر البرنامج والتعليمات أو العبارات ذات الصيغة الأمرية للحاسب وتنفذ هذه الاجراءات حسب تسلسل ورودها و يلاحظ استعمال تعليمة OPEN لغرض فتح الملف مع تحديد نوع الملف INPUT أو OUTPUT ثم تعليمة PERFORM التي تتطلب إنجاز الفقرة READ-WRITE عددا من المرات لغاية تحقق الشرط "YES" = END - OF - JOB ثم ترد تعليمة إغلاق الملف بعد الانتهاء منه CLOSE وتعليمة نهاية البرنامج المنطقية STOP RUN . أما الفقرة المنفذة عددا من المرات فهي تشتمل على عملية قراءة البطاقات المثقبة READ وفي نهاية البطاقات يتم نقل القيمة الحرفية الثابتة YES إلى الحقل END - OF - JOB ثم نقل محتوى البطاقة إلى سجل (أو سطر) الطباعة ثم تعليمة الطباعة أو الكتابة WRITE على الطباعة (ملف الطباعة) بعد التقدم سطر واحد AFTER 1 LINE ثم تعليمة نهاية الفقرة EXIT .

و يلاحظ في قسم الاجراءات أن أسماء الفقرات تبدأ ضمن الأعمدة ٨ - ١١ أما التعليمات ومحتوى الفقرة نفسها فيرد بعد العمود ١٢ ولغاية ٧٢ .

وسوف يتم ذكر تفاصيل أكثر عن هذه الأقسام في الفقرات والفصول التالية .

أما إشارة النجمة (*) في العمود ٧ فهي تعني أن السطر المذكورة فيه يعتبر كسطر ملاحظات ضمن البرنامج ولا يتم تنفيذه .

١ - ٣ الخطوات المنطقية لكتابة برنامج كوبول

يبربرنامج كوبول بعدة مراحل قبل أن يصبح جاهزاً للتشغيل ، هذه المراحل متسلسلة تسلسلاً منطقياً وعملياً ولا يمكن تقديم أو تأخير خطوة عن خطوة وهذه

الخطوات هي :

- ١ - تعيين وتحليل المسألة PROBLEM STATEMENT AND ANALYSIS
- ٢ - رسم مخطط انسيابي للعمل FLOWCHARTING
- ٣ - ترميز البرنامج CODING
- ٤ - تمرير البرنامج على المترجم COMPILER
- ٥ - تنفيذ البرنامج بعد الترجمة EXECUTION
- ٦ - توثيق البرنامج DOCUMENTATION

١ - **تعيين وتحليل المسألة PROBLEM STATEMENT AND ANALYSIS**
يجب علينا أولاً تفهم المسألة المعطاة، والمداخل لهذه المسألة وشكل المخرجات المطلوب، وكذلك طريقة المعالجة بالتفصيل، وكلما كان فهم المسألة، أعمق كلما سهل علينا رسم مخطط انسيابي صحيح لها ومتابعة حل البرنامج بصورة سليمة وسريعة.

٢ - **رسم مخطط انسيابي للمسألة (البرنامج) PROGRAM FLOWCHART**
بعد تفهم المسألة (البرنامج) جيداً يمكن أن نبدأ برسم المخطط الانسيابي. وهو عبارة عن أشكال ترمز إلى عمليات معينة - انظر الشكل رقم (١ - ١) - ويمكننا ترتيب هذه الرموز بشكل منطقي حسب معطيات المسألة وحسب الحل الذي نراه مناسباً وكلما كان المخطط الانسيابي جيداً ودقيقاً كلما استطعنا أن نحول هذه الأشكال إلى تعليمات خاصة بلغة كوبرول بشكل سليم. انظر الشكل (١ - ٢).

٣ - **الترميز CODING**
وهذه المرحلة تتضمن تحويل المخطط الانسيابي إلى لغة كوبرول ويقوم بها المبرمج طبعاً، إذ يحول أو يترجم كل خطوة من خطوات هذا المخطط إلى تعليمة كوبرول أو

أكثر، ويجب أن تتم عملية الترميز بدقة فائقة واتباع أصول وقواعد لغة كوبول، وتتم عملية الترميز على ورق خاص يسمى COBOL CODING SHEETS وعند الانتهاء من كتابة البرنامج على هذا الورق يكون البرنامج جاهزاً وبصيغة تسمى صيغة الأصل SOURCE. انظر الشكل رقم (١ - ٣).

٤ - عملية الترجمة COMPILATION

بعد عملية الترميز يُدخل البرنامج للحاسب عن طريق وحدات ادخال مختلفة، اما عن طريق بطاقات مثقبة ومن ثم قارئ البطاقات CARD READER أو عن طريق شريط ممغنط KEY TO TAPE والشكل الشائع حالياً والأسهل استعمالاً عن طريق النهايات TERMINALS باستخدام برنامج منقح النصوص TEXT EDITOR ويتم التخزين إما على قرص ممغنط DISC أو على قرص صغير يسمى DISKETTE.

بعد عملية التخزين هذه يتم تمرير هذا البرنامج على برنامج موجود في الحاسب يدعى البرنامج المترجم COMPILER ويقوم المترجم هذا بتحويل برنامج كوبول من صيغة الأصل SOURCE الذي كنا قد كتبناه في الخطوة رقم ٣ - الترميز - الى صيغة خاصة بالحاسب MACHINE LANGUAGE. ويكون البرنامج عندها بصيغة قابلة للتنفيذ - عند خلوه من الأخطاء القاعدية طبعاً - وتسمى هذه الصيغة OBJECT. ناتج عملية الترجمة COMPILATION هذه يمكن أن يحتوي على أخطاء قاعدية SYNTAX ERRORS. فيجب أن تصحح هذه الأخطاء وتعاد عملية الترجمة COMPILATION من جديد، ولا نستطيع الانتقال للخطوة التالية حتى يكون البرنامج خالياً من الأخطاء FREE OF ERRORS.

(هـ) تكتب أحياتاً DISK

٥ - تنفيذ البرنامج EXECUTION

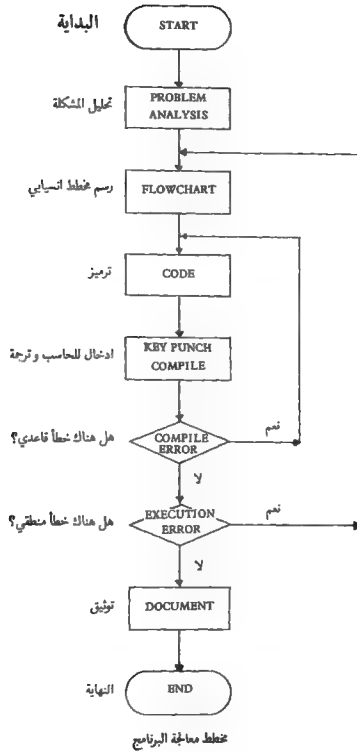
بعد أن يكون البرنامج قد نفع تماماً من الأخطاء القاعدية وتشكل له OBJECT أي صيغة التنفيذ، نستطيع تجربة البرنامج أي عمل TEST أو RUN له. بعد تحضير الملفات الخاصة ان وجدت. بعد عملية التنفيذ هذه ندقق النتائج OUTPUTS، فإذا احتوت على أخطاء نرجع ونعدل أصل البرنامج SOURCE وكذلك المخطط الانسيابي لهذا البرنامج إن لزم الأمر، هذه الأخطاء تسمى الأخطاء المنطقية LOGICAL ERRORS، ثم تعاد عملية الترجمة COMPILATION من جديد، وهكذا حتى نحصل على نتائج سليمة تماماً. عندها فقط يكون البرنامج صحيحاً وجاهزاً للاستعمال.

٦ - التوثيق DOCUMENTATION

إن مرحلة التوثيق تعتبر مرحلة مهمة على المدى البعيد، إذ أنه بعد فترة من الزمن يمكن أن تتغير معطيات المسألة - البرنامج - ونحتاج الى تعديل البرنامج من جديد، ومن الممكن جداً أن يكون المبرمج الذي كتب هذا البرنامج قد نقل الى مكان آخر أو قد بدل عمله أو أي احتمال آخر، لذا لا يشترط بعملية صيانة البرنامج - تعديل البرنامج - أن يقوم بها نفس المبرمج الذي كتبه، لذلك يجب على كل مبرمج بعد الانتهاء من كتابة برنامجه تماماً والتأكد من خلوه من الأخطاء القاعدية والمنطقية أن يحتفظ بكل المعلومات اللازمة عن هذا البرنامج وأن يحتفظ بآخر نسخة صحيحة عن البرنامج ونسخة من المدخلات INPUTS ونسخة من المخرجات OUTPUTS، والمخطط الانسيابي للبرنامج PROGRAM FLOWCHART. يحتفظ بذلك في ملف خاص يسلم لمدير المشروع أو للمسئول عن النظام.

بذلك يكون المبرمج قد كتب برنامجاً متكاملأً بأسلوب صحيح وبالسرية والدقة المطلوبة.

الشكل التالي يوضح تسلسل هذه العمليات :



١ - ٤ العناصر المكونة للغة كوبول

لغة كوبول هي لغة تخاطب بين الانسان والحاسب الآلي مثلها مثل أي لغات لها قواعد في الكتابة وتركيب خاص للجملة لكي تكون مفهومة (معروفة) من قبل مترجم لغة كوبول الذي وضع بناء على هذه القواعد والأسس. وتتكون اللغة من ستة عناصر أساسية وهي :

| | |
|-----------------------------|------------------------------|
| Reserved Words | ١ - الكلمات المحجوزة |
| Programmer - Supplied names | ٢ - أسماء المتغيرات |
| literals | ٣ - الثوابت |
| Symbols | ٤ - الرموز (الإشارات) الخاصة |
| Level numbers | ٥ - أرقام المستوى |
| Picture | ٦ - وصف الحقل (أو المتغير) |

وكل تعليمة ترد في هذه اللغة يجب أن تحتوي على الأقل كلمة محجوزة لتعطي المعنى المراد من هذه التعليمة أو العبارة. وهذه الكلمات يجب أن تذكر دون أدنى خطأ في التهجئة لأنها الأساس الذي يعتمد عليه مترجم اللغة أثناء قراءته للبرنامج. كما أن هذه الكلمات تختلف من مترجم لآخر ويجد القارئ في الملحق (١٥ - ٢) من هذا الكتاب قائمة بهذه الكلمات. كما يجب على المبرمج المبتدئ الانتباه الى هذه الكلمات والرجوع اليها لسببين :

أ - للتأكد من صحة كتابتها.

ب - لتجنب استخدامها كأسماء للحقول أو المتغيرات المراد تسميتها في البرنامج.

أما بالنسبة لأسماء المتغيرات وهي العنصر الثاني فهي عبارة عن مسميات يطلقها المبرمج على الأماكن (الحقول) المخصصة في الذاكرة الرئيسية للحاسب ويستخدمها

(٥) يقصد بالتعليمة كلمة Statement أو Instruction

المبرمج في برنامجه ومعالجته للبيانات الخاصة بها وهذه المسميات تشمل :

- أسماء الفقرات في قسم الاجراءات .
- أسماء أماكن تخزين البيانات (الحقول ، السجلات) .
- أسماء الملفات المستخدمة في البرنامج .

وكل واحد من هذه المسميات يعتبر بمثابة عنوان * لكان خصص في الذاكرة يمكن المبرمج من الاشارة اليه لاستخدام محتواه في المعالجة . مثل PRINT - F تمثل اسم ملف الطباعة في البرنامج المذكور في الفقرة السابقة وكذلك PRINT - LINE اسم سطر الطباعة (سجل الطباعة) و START - P اسم للفقرة في قسم الاجراءات . (الشكل رقم ١ - ٣) .

وعندما يختار المبرمج أسماء متغيرات برنامجه يجب عليه أن يراعي القواعد التالية :

١ - يمكن أن يحتوي الاسم على الأحرف من A الى Z والأرقام من (صفر) ٠ إلى ٩ (تسعة) وإشارة الشرطة (-) ** فقط وبدون أي فراغ بين الأحرف والأرقام المكونة للاسم .

- ٢ - يجب أن يحتوي الاسم على حرف واحد على الأقل ما عدا أسماء الفقرات والفصول *** يمكن أن تكون أرقاماً فقط .
- ٣ - لا يمكن أن يتكون أول رمز أو آخر رمز من الاسم إشارة الشرطة .
- ٤ - لا يمكن استخدام أسماء الكلمات المحجوزة كأسماء متغيرات .
- ٥ - يجب أن لا يزيد حجم الاسم عن ٣٠ رمزاً **** .

(٥) tag

(**) Hyphen

(***) Paragraph and Section names

(****) قد يوجد بعض المترجمات ثقيل حجماً أكبر من هذا الحجم القياسي .

والأمثلة التالية توضح كيفية استخدام هذه القواعد في وضع أسماء المتغيرات :

| اسم المتغير | التوضيح |
|-------------------|-------------------------------------|
| NAME | اسم مقبول |
| DIVISION | اسم غير مقبول لأنه يمثل كلمة محجوزة |
| DIVISION - 1 | اسم مقبول |
| DIVISION - OF - | اسم غير مقبول لوجود شرطة في نهايته |
| DIVISION - OF - X | اسم مقبول |
| 12345 | اسم مقبول لفقرة أو لفصل ولكن |
| | غير مقبول كاسم لمتغير آخر |

ويجب التنويه أن جميع الأسماء تكتب بالأحرف الكبيرة ولا تقبل الأحرف الصغيرة في أسماء المتغيرات .

أما المنصر الثالث في اللغة وهو الثوابت . و يقصد بالثابت قيمة محددة حرفياً أو ثابتاً رقمياً . وعلى ذلك فإنه يمكن أن يكون رقمي (عدي) مثل 1983 أو 261.56 أو 10 .

أما إذا كان الثابت يمثل قيمة غير رقمية فإنه يظهر في لغة كوبرول بين إشارتين من الإشارة * (الفاصلة العلوية) * * .

وتعتبر هذه القيمة ثابتة في البرنامج ويمكن معالجتها أو مقارنتها مع حقول أخرى .
مثل :

MOVE 'INSTITUTE OF PUBLIC ADMINISTRATION' TO PRINT - LINE.

(*) Crower and Crawford P. 22

(**) بعض المترجمات تستخدم الإشارة " بدلا " (الفاصلة العلوية) .

ووفق قواعد لغة كوبول لا يمكن أن يزيد حجم الثابت الرقمي على ١٨ رقماً ويمكن أن تبدأ من جهة اليسار بإشارة جبرية (سالبة أو موجبة) كما يمكن أن يحتوي الثابت على فاصلة عشرية ولكن يجب أن لا يكون آخر رمزه فيه الفاصلة العشرية .

والمعاصر الرابع في مكونات لغة كوبول هو العلامات الخاصة ويمكن تصنيفها الى ثلاثة أنواع :

| | |
|-------------|------------------------------|
| Punctuation | - علامات تنقيط |
| Arithmetic | - علامات حسابية |
| Relational | - علامات العلاقة أو المقارنة |

و يوضح الشكل رقم (١ - ٤) هذه العلامات والغرض منها .

وتستخدم الفاصلة المنقوطة من أجل سهولة قراءة برنامج كوبول وحذفهما سوف لا يؤدي الى خطأ في الكتابة .

| النوع | العلامة | الغرض منها |
|----------------------|---------|--|
| تنقيط Punctuation | . | تشير الى نهاية أي عبارة أو جملة في لغة كوبول |
| | , | لفصل بين العبارات في جملة كوبول |
| | ; | |
| | ' أو '' | لتوصيف الثوابت غير الرقمية |
| | () | لاحتواء المؤشرات أو الأدلة أو التعميمات الحسابية |
| حسابية Arithmetic | + | للجمع |
| | - | للطرح |
| | * | للمضرب |
| | / | للتقسيم |

| الفرص منها | العلامة | النوع |
|-----------------|---------|-------------------------------|
| للرفع الى قوة | == | علامة أو مقارنة Relational |
| مقارنة المساواة | = | |
| لمقارنة أكبر من | > | |
| لمقارنة أصغر من | < | |

الشكل رقم (١ - ٤)
العلامات الخاصة في لغة كوبرول

أما النقطة Period فذات أهمية كبيرة في كتابة جل وعبارات لغة كوبرول وحذفها قد يؤدي الى حدوث أخطاء كبيرة منطقية وشكلية. وعلامات التنقيط تخضع الى قاعدتين أثناء ترجمة برنامج كوبرول :

- ضرورة مطلقة لوجود علامة تنقيط (مثل النقطة أحياناً) وعدم وجودها يؤدي الى حدوث خطأ أثناء الترجمة.
- ضرورة غير مطلقة لوجود علامة تنقيط وعدم وجود هذه العلامة لا يؤدي الى حدوث خطأ في الترجمة بل يقود الى خطأ في تنفيذ البرنامج (خطأ منطقي).
- مثل وجوب وجود فراغ تالي لأية فاصلة أو فاصلة منقوطة أو نقطة وكذلك يجب إنهاء جميع عبارات أو جل لغة كوبرول Entries بنقطة.

أمثلة :

1. INPUT-OUTPUT SECTION
2. OPEN INPUT CARD-FILE,M-FILE.
3. OPEN OUTPUT PRINT-F.

ففي 1 و 2 جملة غير مقبولة الأولى لعدم وجود نقطة في نهاية العبارة والثانية لعدم وجود فراغ بعد الفاصلة.

وسوف يتم شرح العلامات الحسابية والملائقية بالتفصيل في الجزء التالي من الكتاب.

أما العنصر الخامس من لغة كويول فهو أرقام المستوى و يقصد بها الأرقام التي توضح علاقة الحقول الجزئية بالحقول الاجمالية وكيفية تجزئتها . فمثلا وصف سجل بطاقة ما يأخذ الشكل التالي :

φ1 CARD-REC.

φ2 CARD-NAME PICTURE IS A(25).

φ2 CARD-NUMBER PICTURE IS 9(5).

φ2 CARD-ADDRS.

φ5 CARD-CITY PICTURE IS A(12).

φ5 CARD-STREET PICTURE IS X(12).

φ2 FILLER PICTURE IS X(16).

والرقم φ1 يعني رقم المستوى الأول و يعني أن CARD-REC اسم الحقل (مكان التخزين) اجمالي يحتوي على حقول جزئية تأخذ رقم المستوى φ2 ، φ5 ، ... الى غاية الرقم 49 وكذلك الحقول الموصوفة بالرقم φ5 هي حقول جزئية للحقل CARD-ADDRS'.

وسوف يأتي شرح أكثر تفصيلاً لذلك في الفصول القادمة .

أما العنصر السادس وهو وصف الحقل بعبارة PICTURE التي تستخدم لتحديد شكل الحقل وطبيعة البيانات المحتواة فيه سواء كان مخصصاً لغرض ادخال هذه البيانات أو اخراجها وسوف نأتي لذكر هذه العبارة بصورة أكثر تفصيلاً فيما بعد ، ولكن يمكن ذكر ثلاثة أنواع أساسية لوصف الحقل وتحديد نوع محتواه وهي الرمز 9

(هـ) وترجم بالعربية الى صورة (أي شكل الحقل وطريقة توصيفه) .

ويعني أن الحقل الموصوف يأخذ شكل رقمياً أما الرمز A فيعني أن الحقل أبجدي الشكل. أما الرمز X فيدل على أن الحقل يمكن أن يحتوي على كلا النوعين، أبجدي ورقمي مع امكانية احتوائه للإشارات الخاصة.

١ - ٥ صفحة ترميز برنامج كوبول COBOL CODING FORM

يتم كتابة تعليمات وعبارات لغة كوبول على ورق معين (خاص) صمم وفقاً لقواعد كتابة كوبول وفق الشكل رقم (١ - ٥) و يبلغ طول السطر في الصفحة (٨٠) عموداً أو موضعاً لكتابة تعليمات وجمل لغة كوبول ويحول كل سطر في هذه الصفحة الى بطاقة مثقبة في حال ادخال البرنامج بواسطة البطاقات المثقبة أما عند استخدام منقح النصوص TEXT EDITOR فإن كل سطر في الصفحة يعادل سطراً واحداً على الشاشة المرئية.

وفي أعلى الصفحة يوجد مكان مخصص لكتابة الاشياء التعريفية العامة بالبرنامج والنظام واسم المبرمج وتاريخ كتابة البرنامج واسم النظام واسم البرنامج وتعليمات إدخال (أو تثقيب) البرنامج الخاصة وشكل البطاقة المراد تثقيبها (يوجد نوعين من أشكال التثقيب A و B معروفين عالمياً). وكذلك رقم الصفحة في أقصى اليمين من الورقة.

وتعليمات إدخال أو تثقيب البطاقات الخاصة بالبرنامج خاصة عندما يطلب المبرمج من عامل إدخال البيانات القيام بهذه المهمة. ولكي يتجنب المبرمج أي خطأ في عملية الإدخال عليه أن يوضح للشخص الآخر الحروف والأرقام التي يوجد بينها تشابه وأن يميز بينها : مثل الحرف o والرقم ٥ صفر الذي يوضع عليه الإشارة / وكذلك الحرف I والرقم 1 والحرف S والرقم 5. فيمكن للمبرمج توضيح ذلك في الجزء العلوي من الصفحة المخصص لهذا الغرض.

ولتسهيل مهمة عامل ادخال البيانات يكتب عادة البرنامج بشكل يتخلله أسطر فارغة (سطين كتابة بينها سطر فارغ) .

والجزء المهم في صفحة الترميز هو الذي يلي الجزء العلوي حيث أن كل سطر مقسم الى (٨٠) عمود وكل عمود يقابل عمود في البطاقة المثقبة . وتوجد أرقام لهذه الأعمدة في أعلى الصفحة وفي أسفلها لتسهيل احتساب عدد الأعمدة في السطر على المبرمج . وكل عمود (او موضع) مخصص لحرف أو رقم أو فراغ أو اشارة خاصة .

وننتقل الآن الى السطر الواحد حيث أن الستة المواضع (الأعمدة) الأولى مخصصة للرقم المتسلسل للسطر وقد كانت أهمية هذا الرقم عالية جداً عند استخدام البطاقات المثقبة لادخال البرنامج بينما انخفضت أهميته بعد استخدام منقح النصوص لتسجيل البرنامج . حيث أنه يحفظ تسلسل الأسطر (البطاقات) وفق التسلسل المنطقي الموضوع للبرنامج ، وعادة هذا الرقم يقسم الى جزئين كل جزء مؤلف من ثلاثة أعمدة الأول من العمود ١ الى ٣ مخصص لرقم الصفحة والثاني من العمود ٤ الى ٦ مخصص لرقم السطر ضمن الصفحة .

أما العمود رقم ٧ والذي يحمل العنوان Cont فانه يستخدم للاشارة الى أن بعض تعليمات كوبول تزيد عن السطر المخصص وتحتاج الى تمة (Continuation) وتستعمل الاشارة (ـ) للدلالة على ذلك . أما اذا وضعت الاشارة (e) في هذا العمود فان ذلك يعني أن هذا السطر مخصص لاثبات ملاحظات وتعليق عن البرنامج وليس سطراً من ضمن أسطر التعليمات ولا يقوم الحاسب بتنفيذه أثناء تنفيذ البرنامج .

والجزء الفعلي المخصص لكتابة تعليمات وعبارات كوبول في السطر يكون من العمود رقم ٨ الى العمود رقم ٧٢ وهذه المساحة مقسومة الى قسمين A و B الاعمدة ٨ الى ١١ والاعمدة ١٢ الى ٧٢ على التوالي .

والسبب في هذا التقسيم الى جزئين A و B واضح وبسيط وذلك من أجل سهولة القراءة وتلوين البرنامج حيث أن النقاط الرئيسية والتي تعتبر علامة بارزة في البرنامج تبدأ من الجزء A وتدعى (KEY LINES) أما الأسطر التي تعتبر أقل أهمية فتبدأ في الجزء B. فمثلاً أسماء أقسام البرنامج والفصول التابعة لها تبدأ من الجزء A وكذلك أرقام ومستويات توصيف البيانات. وفي قسم الاجراءات تكتب كافة أسماء الفقرات (PARAGRAPHS) ابتداءً من الجزء A. أما بقية الأسطر والتعليمات فتبدأ من الجزء B.

وأخيراً يوجد جزء معنون باسم (IDENTIFICATION) ويأخذ الأعمدة ٧٣ الى ٨٠ من السطر فهو مخصص لاضافة معلومات اختيارية عن هوية السطر وعادة يدون فيها أحرف خاصة باسم المبرمج أو البرنامج مع أرقام متسلسلة أو أرقام مميزة للبرنامج أو الفقرات ضمن البرنامج. وهذا الجزء حسب قواعد لغة كوبول لا يُأخذ بعين الاعتبار أثناء ترجمة (COMPILATION) البرنامج الى لغة الهدف (OBJECT) « او صيغة التنفيذ.

إن وجود صفحة خاصة لترميز برنامج كوبول لا يعني عدم إمكانية كتابة تعليمات وعبارات كوبول على ورق أو صفحات عادية ولكن شكل هذه الصفحة يسهل الكتابة وفق القواعد الموضوعة للغة. وظهور الشاشة المرئية واستخدام منقح النصوص قلل من أهمية استخدام هذا الشكل المخصص للكتابة، ولكنه لا زال استخدامه أمراً ضرورياً خاصة في مراحل تعليم لغة كوبول الأولية.

١-٦ تمارين

- ١ - بين فيما اذا كانت هذه العبارات صحيحة أم خطأ؟
 - الثوابت غير الرقمية يمكن أن لا تحتوي أرقاماً .
 - استعمال الأعمدة ٧٣ الى ٨٠ في صفحة الترميز اختياري .
 - تظهر أسماء الملفات في ثلاثة أقسام DIVISIONS .
 - العمود رقم ٨ يستخدم للدلالة على استمرارية التعليمة أو الجملة .
 - الثابت الرقمي يمكن أن يحتوي على ١٨ رمزاً .
 - يمكن أن تظهر الكلمات المحجوزة Reserved Words في الثوابت غير الرقمية .
- ٢ - بين أي من هذه الأسماء الموضوعية لتسمية الحقول في البرنامج مقبولة وصحيحة وإذا كان الاسم غير مقبول وضح سبب ذلك :

- a) NUMBER - OF - STUDENTS
- b) 1 2 3 4 5 6
- c) FIVE SIX
- d) IDENTIFICATION - DIVISION
- e) STORE -

- ٣ - أشر طبقاً لقواعد كتابة لغة كوبرول أي من هذه التعليمات يعتبر مقبولاً وصحيحاً وأي منها خطأ :

- a) OPEN INPUT R-FILE, Z-FILE.
- b) MOVE B TO C.
- c) MOVESPACE TO T1, T2.

- d) MOVE 'A' TO X.
- e) WRITE RECORD - IN AFTER TWO LINES.
- f) CLOSE IN-FILE/~~PRINT-FILE~~.
- g) CLOSE ~~CARD-FILE~~ PRINT-FILE.

الفصل الثاني

التعليمات الأولية في لغة كوبول

(قسم الاجراءات)

٢ - ١ ملحة عامة

إن قسم الاجراءات PROCEDURE DIVISION في برنامج لغة كوبول هو الجزء الذي يحتوي على المنطق في البرنامج، وهو القسم الذي يقوم عملياً بأداء الغرض من البرنامج، وسوف نحاول تغطية التعليمات الأولية والمهمة في لغة كوبول ضمن هذا الفصل. وتشمل هذه التعليمات عمليات إدخال وإخراج البيانات من وإلى البرنامج (أي الحاسب الآلي) وهي: OPEN, READ, WRITE, CLOSE, وكذلك التعليمات التي تقوم بأداء العمليات الحسابية ADD, SUBTRACT, MULTIPLY, DIVIDE, COMPUTE وتعليمات MOVE التي تنقل البيانات من المناطق المختلفة ضمن الذاكرة الرئيسية في الحاسب وتعليمات المقارنة IF التي تمكن المبرمج من الاختيار بين أكثر من اتجاه في سير البرنامج وتعليمات PERFORM التي تنفذ عملية الدوارة ثم تعليمات إنهاء البرنامج بصورة منطقية وهي . STOP RUN

٢ - ٢ قراءة صيغ لغة كوبول

لقد تم وضع أشكال عامة لمجموعة صيغ لغة كوبول ويمكن قراءة هذه الأشكال وفق الأسس التالية :

(*) LOOP

- ١ - تظهر الكلمات المحجوزة ضمن الشكل العام بالأحرف الانكليزية الكبيرة
- ٢ - يوضع خط تحت الكلمات المحجوزة التي يجب أن توجد في العبارة أو الجملة ، أما الكلمات الاختيارية Optional فلا يوضع خط تحتها .
- ٣ - تشير الكلمات المكتوبة بالأحرف الصغيرة الى أسماء المتغيرات التي تعطى من قبل المبرمج .
- ٤ - تشير الأقواس المربعة [] الى العبارات أو الكلمات الاختيارية في التعليمة أو الجملة .
- ٥ - تشير الأقواس { } إلى أن أحد البنود المذكورة ضمنها الذي يجب اختياره أو ذكره في العبارة .
- ٦ - تشير النقاط (...) الى امكانية تكرار المقطع الأخير من العبارة عددا من المرات ٥ . ويمكن توضيح هذه العلامات أو الرموز في المثال التالي الذي يوصف جزءا من تعليمة IF :

$$\text{IF} \left\{ \begin{array}{l} \text{identifier - 1} \\ \text{literal - 1} \end{array} \right\} \left\{ \begin{array}{l} \text{IS [NOT] GREATER THAN} \\ \text{IS [NOT] LESS THAN} \\ \text{IS [NOT] EQUAL TO} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier - 2} \\ \text{literal - 2} \end{array} \right\}$$

فكلمة IF اجبارية في التعليمة لأنه يوجد خط تحتها وكذلك الكلمات NOT, EQUAL, GREATER, LESS ، أما كلمة Identifier - 1 وكلمة Literal - 1 فيذكر أحدهما في التعليمة ، وهما تمثلان الاسم أو القيمة التي يحددها المبرمج . بينما الكلمات IS, THEN, TO فهي اختيارية في التعليمة هذه .

(٥) عدد مرات تكرار أي مقطع تحكمي Arbitrary و يضبطة نوع المترجم المستخدم و يذكر في مراجع الشركة الصانعة للحاسب عادة .

٢- ٣ التعليمات الأولية للمعالجة ونقل التسلسل :

وتشمل هذه التعليمات ما يلي :

- تعليمات القراءة والكتابة وفتح وغلق الملف :

OPEN, READ, WRITE, CLOSE

- تعليمة نقل وتحريك البيانات :

MOVE

- تعليمات نقل تسلسل تنفيذ البرنامج والمقارنة والتوقف

GO TO, PERFORM, IF, STOP

٢- ٣- ١ تعليمة فتح الملف OPEN

كل ملف في برنامج كوبول يراد معالجته يجب ذكر تعليمة OPEN قبل هذه المعالجة. وتطلب هذه التعليمة من نظام التشغيل توفير الملف المعروف على الوحدات المتصلة بالخاصب سواء كان هذا الملف مخصصا للقراءة أو للكتابة (التسجيل) عليه. وتأخذ هذه التعليمة الشكل العام التالي :

$$\text{OPEN} \left\{ \begin{array}{l} \text{INPUT} \\ \text{OUTPUT} \\ \text{I - O} \\ \text{EXTEND} \end{array} \right\} \text{file - name - 1 [,file - name - 2]}$$

وضمن هذه التعليمة يحدد المبرمج الغرض من استخدام الملف في البرنامج فكلمة INPUT تعني أن الملف تم فتحه لغرض القراءة أو لإدخال البيانات منه للبرنامج وكلمة OUTPUT تعني أن الملف سوف يستخدم لغرض التسجيل عليه، أما كلمة

I - O فهي تستخدم لوصف الملف لأجل عمليات القراءة والكتابة معاً في نفس البرنامج أما كلمة EXTEND وتعني توسيع حجم الملف وتستعمل لغرض الاضافة على ملف سابق سجل على شريط مخنط بشكل خاص . ويمكن لتعليمة OPEN أن تحتوي أكثر من نوع بنفس الوقت لاكثر من ملف .

أمثلة :

1. OPEN INPUT CARD-FILE.
2. OPEN INPUT CARD-FILE, OUTPUT PRINT-F .
3. OPEN I - O SCRATCH - F.
4. OPEN EXTEND MTP-FILE.

٢-٣-٢ تعليمة قراءة البيانات READ

تستخدم تعليمة READ لقراءة البيانات من خارج برنامج كويك (أي من وحدات الإدخال والإخراج المتصلة مباشرة بالحاسب) لكي تتم معالجتها في البرنامج . وتأخذ هذه التعليمة الشكل العام المختصر التالي :

READ file - name [AT END imperative - statement]

حيث أن اسم الملف المراد قراءته يرد بعد كلمة READ وتتم القراءة من الملف على أساس سجل واحد لكل تعليمة قراءة . أما عبارة AT END الاختيارية فهي تمثل شرطاً يستحق عندما تصل عملية القراءة الى نهاية الملف ولم يعد يوجد سجلات لأجل القراءة عندئذ ينتقل تسلسل التنفيذ الى التعليمة الواردة مباشرة بعد كلمة END .

مثال :

READ CARD - FILE AT END MOVE ' NO ' TO MORE - DATA

ففي هذا المثال تقوم تعليمة القراءة بقراءة سجل من الملف المسمى CARD - FILE وعندئذ تصل القراءة (في حال تكرارها من قبل المبرمج في برنامجه) الى نهاية الملف تتم

عملية نقل القيمة الثابتة NO الى الحقل المسمى MORE - DATA . أما إذا لم يتحقق شرط لنهاية الملف فيتابع الحاسب تنفيذ التعليمة التالية لتعليمة READ مباشرة .
ويجب التأكيد على أن تعليمة READ تأخذ اسم الملف المقروء الى جانبها وليس اسم السجل .

٢ - ٣ - ٣ تعليمة نقل البيانات MOVE

تقوم تعليمة النقل MOVE في لغة كوبرول بعملية نقل البيانات (قيمة الحقل) وتحريكها بين أماكن التخزين في الذاكرة، وهذا النقل لا يعني النقل الفيزيائي للشيء المنقول بل هو عبارة عن نسخ للمحتوى ويبقى نفس المحتوى في حقل مصدر البيانات .
وشكل تعليمة النقل العام كما يلي :

$$\text{MOVE} \left\{ \begin{array}{l} \text{Identifier - 1} \\ \text{Literal} \end{array} \right\} \text{TO Identifier - 2 , Identifier - 3]}$$

حيث تعني كلمة Identifier - 1 اسم الحقل أو مكان التخزين في الذاكرة والموصوف من قبل المبرمج في قسم البيانات . وكلمة Literal تعني القيمة الحرفية أو الثابتة المراد نقلها (أو بالأحرى نسخها) .

أمثلة :

- 1) MOVE 205 TO P - O - BOX.
- 2) MOVE 'RIYADH' TO CITY - NAME.
- 3) MOVE STUDENT - NAME TO PRINT - NAME.

ففي المثال الأول تقوم تعليمة النقل بوضع القيمة 205 في الحقل P - O - BOX أما المثال الثاني فتنم عملية نقل كلمة RIYADH الى الحقل CITY - NAME أما الثالث

فيمتضمن نقل محتوى (نسخ محتوى) الحقل NAME - STUDENT الى الحقل مستقبل البيانات وهو PRINT - NAME.

ويمكن التمييز في عملية نقل البيانات MOVE ضمن لغة كوبرول بين نوعين من البيانات :

١ - البيانات غير الرقمية : وتتم عملية النقل (النسخ) بين حقل مصدر البيانات SOURCE والحقل مستقبل البيانات RECEIVING على أساس رمز واحد من البيانات في كل مرة وعلى التوالي من جهة اليسار الى اليمين . وإذا كان حقل مستقبل البيانات أكبر من حقل المصدر يُملأ الجزء الفائض من هذا الحقل بعد عملية تنفيذ النقل بفراغات BLANKS أما إذا كان الحقل المستقبل أصغر حجماً من حقل المصدر فان الجزء الزائد من جهة اليمين يُحذف TRUNCATED ولا تتم عملية نسخه :

MOVE A TO B.

مثال :

A :

| | | | | |
|---|---|---|---|---|
| A | B | C | D | E |
|---|---|---|---|---|

B :

| | | | | | |
|---|---|---|---|---|---|
| ? | ? | ? | ? | ? | ? |
|---|---|---|---|---|---|

 قبل التنفيذ :

A :

| | | | | |
|---|---|---|---|---|
| A | B | C | D | E |
|---|---|---|---|---|

B :

| | | | | | | |
|---|---|---|---|---|--|--|
| A | B | C | D | E | | |
|---|---|---|---|---|--|--|

 بعد التنفيذ :

MOVE C TO D.

C :

| | | | | | |
|---|---|---|---|---|---|
| A | B | C | D | E | F |
|---|---|---|---|---|---|

D :

| | | | |
|---|---|---|---|
| ? | ? | ? | ? |
|---|---|---|---|

 قبل التنفيذ :

C :

| | | | | | |
|---|---|---|---|---|---|
| A | B | C | D | E | F |
|---|---|---|---|---|---|

D :

| | | | |
|---|---|---|---|
| A | B | C | D |
|---|---|---|---|

 بعد التنفيذ :

٢ - البيانات الرقمية : تتم عملية النسخ هنا من جهة اليمين الى اليسار وتتم رقم رقم digit في حال عدم وجود فاصل عشرية في الحقل المنقول (المنسوخ) . أما اذا كان الحقل يحتوي على قيمة رقمية عشرية فتكون الفاصلة العشرية هي بمثابة النقطة المرشدة في كلا الحقلين المصدر والمستقبل للبيانات .

وإذا كان حقل مستقبل البيانات أكبر من حقل مصدر البيانات فتم عملية إضافة أصفار إلى يسار الرقم المنقول (المنسوخ) لملء الجزء الفائض بعد النقل. أما إذا كان العكس فيحدث حذف للجزء الزائد عن حجم الحقل المستقبل من جهة اليسار (أي القيم العليا والكبيرة في الحقل الرقمي أو القيمة الثابتة المراد نقلها).

مثال :

MOVE A TO B

لدينا التعليمة :

A :

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

B :

| | | | | | | |
|---|---|---|---|---|---|---|
| ? | ? | ? | ? | ? | ? | ? |
|---|---|---|---|---|---|---|

 قبل التنفيذ :

A :

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

B :

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|

 بعد التنفيذ :

أما إذا كانت التعليمة تعالج حقولاً عشرية فيكون مثلاً

MOVE N TO Q.

حيث أن :

عشري : صحيح

عشري : صحيح

N :

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|

Q :

| | | | | |
|---|---|---|---|---|
| ? | ? | ? | ? | ? |
|---|---|---|---|---|

 قبل التنفيذ :

عشري : صحيح

عشري : صحيح

N :

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|

Q :

| | | | | |
|---|---|---|---|---|
| 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|

 بعد التنفيذ :

وتجدر الإشارة هنا إلى أن الفاصلة العشرية Decimal Point لا تمثل مكاناً خاصاً (Byte) بها وإنما يشار إليها بواسطة إشارة خاصة في الذاكرة و يرمز لها عند تمثيل الحقل بالرمز (Δ).

(e) سوف يرد في الفصل القادم كيفية توصيف الفاصلة العشرية وتفاصيل أخرى عنها.

كما يجب التنويه أيضاً الى أن حجم الحقول الرقمية في عملية المناقلة (MOVE) يجب أن لا يزيد عن ١٨ رمزاً لكل حقل مصدر بيانات بما فيها الإشارة الجبرية. أما الحقول غير الرقمية فيمكن أن تصل الى حجم كبير ويختلف باختلاف الحاسب المستخدم لتنفيذ برنامج كربول.

٢-٣-٤ تعليمية كتابة السجل WRITE

تستخدم تعليمية الكتابة WRITE لكتابة أو تسجيل سجل ما على ملف الاخراج بواسطة وحدات الاخراج، وتأخذ هذه التعليمية الشكل العام التالي :

$$\text{WRITE record - name} \left\{ \begin{array}{c} \text{AFTER} \\ \text{BEFORE} \end{array} \right\} \text{ADVANCING} \left\{ \begin{array}{c} \text{identifier} \\ \text{integer} \end{array} \right\} \left\{ \begin{array}{c} \text{LINE} \\ \text{LINES} \end{array} \right\}$$

حيث أن عبارة ADVANCING اختيارية وتقوم بوظيفة ضبط عدد الاسطر عند تسجيل السجلات (الاسطر) على الطابعة للملف اخراج النتائج من البرنامج. وإذا حذفت هذه العبارة فتتقدم الطابعة سطراً واحداً فقط. وكلمة AFTER تعني الكتابة بعد تقدم عدد من الأسطر.

مثال :

WRITE P - LINE AFTER 3 LINES

ففي هذه التعليمية تتم كتابة السجل P - LINE بعد تقدم الطابعة سطرين فارغين وتتم الكتابة على السطر الثالث. أما كلمة BEFORE فتستخدم لكتابة السجل أو السطر قبل تقدم الطابعة عدد الأسطر المذكور في هذه التعليمية.

ويجب التأكيد هنا الى أن تعليمة WRITE تستخدم اسم السجل المكرر في الملف سواء كان هذا الملف يكتب على وسيط مغناطيسي أم على طابعة . بينما تستخدم تعليمة القراءة READ اسم الملف من أجل قراءة سجل واحد من هذا الملف .

٢-٣-٥ تعليمة غلق الملف CLOSE

يجب غلق الملفات المستخدمة في برنامج كويول بعد الانتهاء من معالجتها وتقوم تعليمة CLOSE بهذه الوظيفة وهذه التعليمة ضرورية خاصة عندما يكون الملف المستخدم ملف اخراج لأنه بعد تنفيذ عملية غلق الملف يضمن المبرمج سلامة السجلات في الملف . وتأخذ هذه التعليمة الشكل العام التالي :

CLOSE file - name - 1 [file - name - 2],...

حيث أن file - name - 1 تمثل اسم الملف المستخدم في البرنامج (عادة يكون الاسم المنطقي للملف) . و يلاحظ أن هذه التعليمة تقوم بعملية غلق ملف أو أكثر ولا تحتوي على نوع الملف (مثل I - O , INPUT , OUTPUT) .

مثال :

CLOSE CARD - FILE.

CLOSE CARD - FILE, PRINT - FILE.

وتجب الإشارة الى أن تعليمة غلق الملف ليس بالضرورة أن ترد في آخر البرنامج بل يمكن أن تكون في أي مكان وحسب وصفها المنطقي والحاجة اليها ، كما يمكن إعادة فتح نفس الملف بعد غلقه بهذه التعليمة بوساطة تعليمة OPEN تالية لها .

٢-٣-٦ تعليمية نقل التسلسل GO TO

تتكون تعليمية «اذهب الى» GO TO المبرمج من نقل تسلسل تنفيذ البرنامج في قسم الاجراءات الى فقرة أخرى. وتعطيه مرونة في التنقل واختيار التعليمات المراد تنفيذها وفق تسلسل البرنامج المنطقي. وتأخذ هذه التعليمية الشكل العام التالي :

GO TO [Procedure - name]

حيث يقصد باسم الاجراء Procedure - name اسم الفقرة المراد القفز اليها بعد تنفيذ GO TO .

مثال :

GO TO READ - RECORD.

GO TO NEXT - PARAGRAPH.

حيث أن READ - RECORD, NEXT - PARAGRAPH, أسماء فقرات (أو اجراءات) معروفة في قسم الاجراءات ضمن البرنامج.

ويجب التنويه هنا الى أن اسلوب البرمجة الحديث يتجه نحو التقليل ما أمكن من هذه التعليمية ضمن البرنامج لأنها من السهل أن تكون سبباً في حدوث اخطاء منطقية بسبب عملية النقل التي تؤديها بين الفقرات ووجودها الكثير ضمن البرنامج يجعله متشابكاً جداً مما يصعب تتبعه وتدقيقه.

٢-٣-٧ تعليمية التوقف STOP

كل برنامج كوبرول يجب أن يحتوي على تعليمية توقف واحدة على الأقل و تتدل هذه التعليمية على النهاية المنطقية لتنفيذها. وتأخذ هذه التعليمية الشكل العام :

(٥) Structured Programming

(٥٥) إذا لم يمكن هذا البرنامج معرف كبرنامج فرعي Subprogram

$$\frac{\text{STOP}}{\quad} \left\{ \begin{array}{c} \text{RUN} \\ \text{literal} \end{array} \right\}$$

حيث أن كلمة RUN عندما ترد مع كلمة STOP تؤدي الى توقف البرنامج عن التنفيذ وتدل على نهايته المنطقية و ينتقل نظام التشغيل بعد ذلك لتنفيذ عمل آخر.

أما اختيار اضافة ثابت حرفي Literal الى جانب كلمة STOP فيتيح إظهار رسالة خاصة لمشغل الحاسب ليقوم بالاجابة بعد اطلاعه عليها . وبذلك تعطي هذه التعليمات عندئذ امكانية تدخل مشغل الحاسب عند الطلب ومعنى آخر تؤدي هذه التعليمات في تلك الحالة الى توقف مؤقت PAUSE ريثما يعطي المشغل اجابته على الرسالة المظهرة . كما تجب الإشارة الى أن هذه التعليمات ليس بالضرورة أن تكون آخر تعليمات في تسلسل البرنامج بل توضع عند النهاية المنطقية للبرنامج ضمن قسم الاجراءات .

مثال :

STOP RUN.

STOP 'IF NO MORE TAPES ENTER 1'.

٢ - ٣ - ٨ - تعليمات نفذ (أنجز) PERFORM

تعتبر تعليمات PERFORM من التعليمات الأكثر أهمية وفعالية في لغة الكوبول . فهي تمكن المبرمج من نقل تسلسل تنفيذ البرنامج من فقرة (Paragraph) إلى فقرة أخرى أو أي فقرة في مكان ما ضمن البرنامج . وتكمن أهمية هذه التعليمات في امكانية تقسيم أي برنامج كوبول متشابك أو معقد الى أجزاء سلسلة وسهلة وواضحة بحيث يمكن صياغة الجزء الرئيسي من البرنامج مشتملاً على عدة تعليمات PERFORM متسلسلة

وكل تعليمة منها تؤدي أو تنجز فقرة أو مجموعة فقرات من بقية البرنامج (أي الأجزاء الفرعية).

وتبرز أهمية استخدام هذه التعليمة عند اللجوء الى تطبيق البرمجة الهيكلية (Structured Programming) في كتابة برنامج كوبرول، ومنبجث ذلك في حينه. وتعليمة نفذ تأخذ أشكالاً مختلفة عامة وأبسطها الشكل التالي :

PERFORM Procedure - name

وعبارة Procedure - name يقصد بها اسم "الأجراء" في قسم الاجراءات أو اسم فقرة من هذا القسم. وفعل التعليمة يطلب من الحاسب تنفيذ هذا الاجراء عند وصول تسلسل تنفيذ تعليمات البرنامج الى هذه التعليمة.

مثال :

OPEN OUTPUT PRINT - FILE, INPUT IN - FILE.

PERFORM WRITE - HEADER - LINES. → (1)

(2) →

READ - NXT.

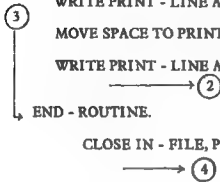
READ IN - FILE AT END PERFORM END - ROUTINE,

→ (3) (4) → STOP RUN.

(1) MOVE FLD - 1 TO OUT - FLD - 1.

GO TO READ - NXT.

WRITE - HEADER - LINES.



وعندما تكون الفقرة أو جزء الاجراء المراد تنفيذه في تعليمة نفذ مؤلف من أكثر من فقرة فيكون شكل تعليمة PERFORM العام التالي :

THROUGH

THRU

Procedure - name - 2

وعبارة THRU الاختيارية تسمح بتنفيذ جزء الاجراءات المراد بتعليمه PERFORM من أول فقرة مسماة وهي 1 - name - Procedure الى الفقرة النهائية فيه وهي 2 - name - Procedure (ملاحظة : الفقرة الأخيرة تعليماتها مشمولة في الجزء المطلوب) .

وغالباً ما تكون آخر فقرة مكونة من اسم لفقرة مضافاً اليها كلمة EXIT . وعبارة EXIT هذه هي تعليمه من تعليمات كويول ولا تقوم بتنفيذ أي عمل ولكنها تعبر فقط عن نهاية جزء الاجراء المرغوب (المسمى) في تعليمه PERFORM .

وشكل تعليمه EXIT العام هو :

Paragraph - name. EXIT

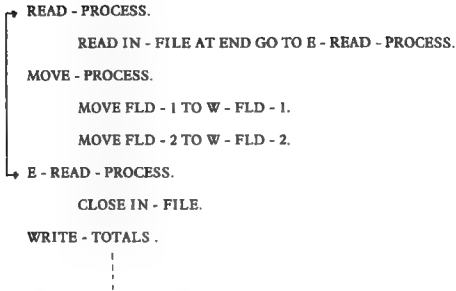
ويجب ملاحظة أن اسم الفقرة السابقة لهذه العبارة ليس إجبارياً فهو اختياري ، لذلك فانه يمكن أن ترد في آخر أي فقرة نهائية للجزء الموصف في تعليمه PERFORM .

وتجدر الملاحظة أنه على المبرمج مراعاة عودة تسلسل تنفيذ البرنامج بعد الجزء المسمى في PERFORM الى النقطة التالية لهذه التعليمه . لهذا السبب يمكن أن تكون عبارة EXIT نقطة إنهاء الجزء المطلوب تنفيذه بتعليمه PERFORM . كما يجب التنويه هنا الى امكانية استخدام عبارة GO TO ضمن الجزء المستدعى في PERFORM على أن لا تسبب نقل التسلسل الى خارج الجزء المحدد لذلك .

مثال :

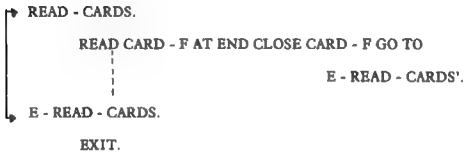
PERFORM READ - PROCESS THRU E - READ - PROCESS.

(هـ) الحاسب T1990 يتطلب اسم فقرة لكل تعليمه EXIT .



مثال :

PERFORM READ - CARDS THRU E - READ - CARDS.



ويسمى المجال ما بين نقطة (فقرة) البداية في الجزء المطلوب ونقطة (فقرة) النهاية

مجال PERFORM أو
- Range of PERFORM -

ويمكن ان تكون مجالات تنفيذ PERFORM متعددة في البرنامج الواحد وتجب عندئذ الإشارة الى تجنب أي تداخل في هذه المجالات لأنه بسبب خطأ منطقياً في سير تنفيذ البرنامج .

مثال صحيح :
عن مجالين متداخلين الأول كبير والثاني أصغر :

- | | | | | |
|---|----|---------|---|---------|
| [| A. | PERFORM | B | THRU F. |
| | B. | _____ | | |
| | C. | PERFORM | D | THRU E. |
| | D. | _____ | | |
| | E. | _____ | | |
| | F. | _____ | | |

مثال صحيح :

عن مجالين متعددين غير متداخلين :

- | | | | | |
|---|----|---------|---|---------|
| [| A. | PERFORM | B | THRU D. |
| | B. | _____ | | |
| | C. | PERFORM | M | THRU Q. |
| | D. | _____ | | |
| [| G. | _____ | | |
| | M. | _____ | | |
| | I. | _____ | | |
| | Q. | _____ | | |

مثال غير صحيح :

بسبب تداخل مجالي تعلية PERFORM

- | | | | | |
|---|----|---------|---|---------|
| [| A. | PERFROM | B | THRU E. |
| | B. | _____ | | |
| | C. | | | |
| | | PERFORM | D | THRU F. |
| | D. | _____ | | |
| | E. | _____ | | |
| | F. | _____ | | |

ولتنفيذ إجراء معين (فقرة أو فقرات معينة) عدد ثابت من المرات تأخذ تعليمة **PERFORM** الشكل التالي :

PERFORM Procedure - name - 1 [THRU Procedure - name - 2]

$\left\{ \begin{array}{c} \text{identifier} \\ \text{literal} \end{array} \right\} \quad \underline{\text{TIMES}}$

حيث يمثل المتغير أو الثابت رقماً صحيحاً يدل على عدد المرات المراد تنفيذها للإجراء المذكور (وتمثل الدوارة Loop). ثم يعود تسلسل تنفيذ البرنامج الى طبيعته بعد الانتهاء من عدد المرات المذكور في تعليمة **PERFORM** والممثل في قيمة المتغير أو الثابت literal . identifier,

مثال :

MOVE 5 TO N.

PERFORM PARA - 1 THRU PARA - 10 N TIMES.

PARA - 1. MOVE ZERO TO TOT - FLD.

PARA - 10. EXIT.

(*) أكبر من الصفر وغير سالب، وفي حال كون هذه القيمة صفراً أو عدداً ثابتاً يتم تنفيذ التعليمة التالية لـ **PERFORM**.

تعليلة نفذ المشروطة :

تأخذ تعليلة PERFORM شكلاً آخر عند استخدام العبارة UNTIL معها :

PERFORM Procedure - name - 1 [THRU Procedure - name - 2]

[UNTIL Condition]

وهذا الشكل العام من التعليلة يضيف شرطاً Condition إليها فإذا تحقق هذا الشرط قبل الوصول الى تنفيذها (تتم المقارنة عليه قبل تعليلة PERFORM) فإن الاجراءات المطلوبة (الفقرات الموصوفة) في التعليلة نفذت سوف لا تتم ولا ينتقل تسلسل البرنامج اليها ، ويتابع الحاسب تنفيذ البرنامج وفق التسلسل العادي له (أي التعليلة التي تلي PERFORM مباشرة) . وإذا لم يتحقق الشرط الموصوف فإن الحاسب ينفذ الاجراءات المطلوبة عدداً من المرات ويتأكد في كل مرة من أن الشرط الموصوف قد تحقق أم لا . وعند تحقق الشرط ينتقل التسلسل الى التعليلة التالية مباشرة .

مثال :

MOVE 10 TO N.

PERFORM P - AA THRU P - EE UNTIL N IS

EQUAL 10. *

وفي هذا المثال فان الحاسب سوف لا ينفذ تعليلة PERFORM لأن الشرط تحقق قبل الوصول اليها (أي أن $N = 10$) .

(*) عبارة $N = 10$ أو عبارة IS EQUAL مستخدمة في لغة الكوبول .

مثال :

MOVE SPACE TO PRINT - LINE.

MOVE 1 TO N.

PERFORM PARA - A UNTIL N = 5. *

PARA - A.

WRITE PRINT - LINE AFTER 1.

إن الفقرة PARA - A سوف تنفذ ٤ مرات وليس ٥ مرات وبعد المرة الرابعة تكون قيمة N مساوية للرقم 5. حيث أن مقارنة القيمة N تتم قبل البدء في تنفيذ التعليمة .PERFORM

مثال :

MOVE ' NO ' TO END - OF - FILE - FLD.

PERFORM READ - CARDS UNTIL END - OF - FILE - FLD IS EQUAL ' YES '.

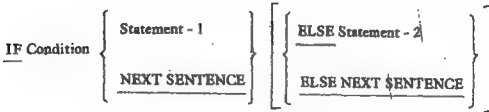
READ - CARDS.

READ CARD - FILE AT END MOVE 'YES' TO END - OF - FILE - FLD.

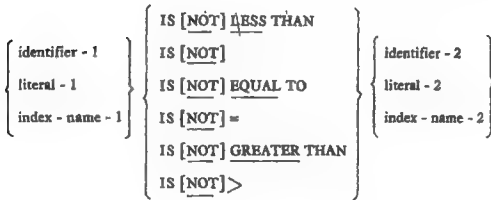
(*) عبارة N = 10 أو عبارة IS EQUAL مستخدمة في لغة الكوبول.

٢-٣-٩ تعليمة المقارنة IF

ان تعليمة المقارنة IF هي التعليمة الأكثر استخداماً في برامج الكويول وذات أثر فعال وكبير في تحويل مسار البرنامج وتأخذ التعليمة الشكل العام التالي :

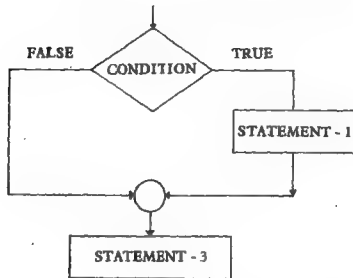


حيث أن كلمة أو عبارة Condition يقصد بها المقارنة المنطقية بين شيئين Relational Condition ويمكن أن تأخذ أحد الأشكال التالية :

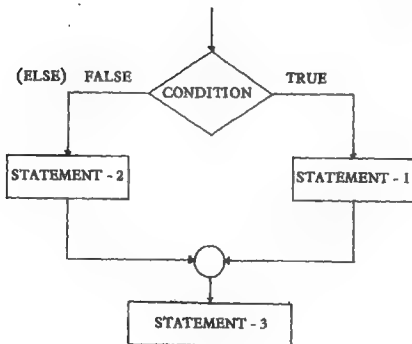


وتعليمة IF يمكن أن تحتوي على عبارة ELSE الاختيارية وعند تحقق الشرط الموصوف بعد كلمة IF ينفذ الحاسب الجزء التالي وهو 1 - Statement وإذا لم يتحقق الشرط ينفذ الحاسب التعليمة التالية لتعليمة IF (أو بالأصح الجملة Sentence التالية لجملة IF لأن IF يجب أن تنتهي بنقطة في آخرها).

والشكل التالي يوضح سير عملية المقارنة واتجاه تنفيذ التعليمة.



رسم لتوضيح تعليمة IF مع عبارة ELSE :



أمثلة :

- 1) IF FLD - 1 IS EQUAL TO 99 GO TO A - END.
- 2) IF VALUE - FLD < 1000 PERFORM MOVE - FLDS
THRU E - MOVE
ELSE DISPLAY "ERROR INPUT"
LINE 21 POSITION 10,
ACCEPT WAIT - FLD.
- 3) IF SUB - CO - E = "1" MOVE SUB - NAME TO FLD - 1 ELSE
MOVE SUB - NAME TO ERR - FLD.

ويجب الإشارة هنا الى أن المقارنة يمكن أن تكون على حقول رقمية أو حقول غير رقمية، ومن حيث المبدأ فإن أية مقارنة تتم رقمياً بوساطة قيمة الأحرف والإشارة الممثلة في الحقل، إلا أنه على المبرمج ملاحظة المقارنة غير الرقمية تتم ابتداء من جهة الشمال في الحقلين محل المقارنة، فمثلاً كلمة BAKER مع كلمة BROWN فنجد BAKER أصغر من BROWN لأن قيمة الحرف B و R أكبر من B و A. و ينصح في مثل هذه المقارنة الرجوع الى مراجع الشركة الصانعة للحاسب للتأكد من قيمة مجموعة الاحرف والاشارات المستخدمة.

أما المقارنة الرقمية فتتم على أساس جبري أي العدد السالب أصغر من الموجب.

٢ - التعليمات الحسابية ARITHMETIC VERBS OR INSTRUCTIONS

في هذا الجزء من لغة كوبرول سنستعرض استخدام التعليمات الحسابية في البرنامج وصيغها المختلفة.

٢-٤-١ تعليمة الجمع ADD

تستخدم تعليمة ADD لأجل انجاز عملية الجمع بأشكالها المختلفة (أي الجمع الى مخزن واحد وتراكم القيمة أو الجمع الى مخزن آخر) وتأخذ تعليمة الجمع شكلين أساسيين * وهما :

$$\begin{aligned}
 1) \quad & \underline{\text{ADD}} \quad \left\{ \begin{array}{c} \text{identifier - 1} \\ \text{literal - 1} \end{array} \right\} \quad \left\{ \begin{array}{c} \text{identifier - 2} \\ \text{literal - 2} \end{array} \right\} \quad \dots \quad \underline{\text{TO}} \quad \text{identifier - n} \\
 & \left[\text{identifier - m} \right] \\
 2) \quad & \underline{\text{ADD}} \quad \left\{ \begin{array}{c} \text{identifier - 1} \\ \text{literal - 1} \end{array} \right\} \quad \left\{ \begin{array}{c} \text{identifier - 2} \\ \text{literal - 2} \end{array} \right\} \quad \left\{ \begin{array}{c} \text{identifier - 3} \\ \text{literal - 3} \end{array} \right\} \\
 & \dots \quad \underline{\text{GIVING}} \quad \text{identifier - m} \quad \left[\text{identifier - n} \right]
 \end{aligned}$$

ففي الشكل الأول يتم جمع قيم المتغيرات (أو الثوابت) 1, 2, ... باضافة النتيجة الى المتغير m أو n الوارد بعد كلمة TO. (أي أن قيمة 1, 2, 3, ... n تخزن في نفس n) بينما الشكل الثاني الذي يحتوي على كلمة GIVING فان عملية الجمع تتم باضافة القيم الواردة في المتغيرات (أو الثوابت) 1, 2, 3, ... ووضع النتيجة في المتغير m أو n الوارد بعد كلمة GIVING دون الأخذ بعين الاعتبار القيمة الموجودة في هذه المتغيرات الأخيرة m, n قبل تنفيذ الجمع.

(*) أغلب الحاسبات تحتوي مترجم لا يقبل الجمع أو الطرح أو الضرب أو القسمة ووضع النتيجة في أكثر من مخزن واحد للنتيجة، لذا يراعى الاطلاع على المترجم المستخدم.

مثال :

ADD A B TO C

| | | | | | | |
|---|---|---|----|---|----|-------------|
| A | 7 | B | 13 | C | 15 | قبل التنفيذ |
| A | 7 | B | 13 | C | 35 | بعد التنفيذ |

فالنسبة 35 تمثل قيمة جمع المتغير A والمتغير B والمتغير C ووضع القيمة النهائية في المتغير C.

مثال :

ADD A B GIVING C.

| | | | | | | |
|---|---|---|----|---|----|-------------|
| A | 7 | B | 13 | C | 15 | قبل التنفيذ |
| A | 7 | B | 13 | C | 20 | بعد التنفيذ |

وهنا تستبدل القيمة الأولية للمتغير C بنتيجة (أو حاصل) جمع المتغير A و B والتي تعادل 20، والجدول التالي يلخص تعليمة الجمع في أمثلة :

ADD تعليمة الجمع

| اسم الحقل : | A | B | C |
|---------------------------------|----|----|----|
| قيمة الحقل قبل التنفيذ : | 15 | 20 | 25 |
| القيمة بعد تنفيذ أي من | | | |
| العمليات الواردة في التعليمات : | | | |
| ADD A TO C. | 15 | 20 | 40 |
| ADD A B TO C | 15 | 20 | 60 |
| ADD A 13 B GIVING C. | 15 | 20 | 48 |
| ADD A 13 B TO C. | 15 | 20 | 73 |
| ADD 1 TO B, C. | 15 | 21 | 26 |

٢-٤-٢ تعليمية الطرح SUBTRACT

تستخدم تعليمية الطرح SUBTRACT لأجل حساب تفاضل قيمتين أو أكثر وحساب الفرق بينهما. وتأخذ الشكلين الأساسيين العاملين :

$$1) \quad \underline{\text{SUBTRACT}} \quad \left\{ \begin{array}{l} \text{identifier - 1} \\ \text{literal - 1} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier - 2} \\ \text{literal - 2} \end{array} \right\} \dots \dots$$

FROM identifier - m identifier - n

$$2) \quad \underline{\text{SUBTRACT}} \quad \left\{ \begin{array}{l} \text{identifier - 1} \\ \text{literal - 1} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier - 2} \\ \text{literal - 2} \end{array} \right\} \dots \dots$$

FROM $\left\{ \begin{array}{l} \text{identifier - m} \\ \text{literal - m} \end{array} \right\}$ GIVING identifier - n

$\left[\text{identifier - 0} \right] \dots$

ففي الشكل (١) فإن قيمة المتغير (الحقل) m تستبدل بقيمة حاصل الطرح (نتيجة الطرح = الحقل m ناقصاً الحقل ١ و ٢).

أما في الشكل (٢) فإن قيمة الحقل (m) أو الثابت (m) لا تتغير في عملية الطرح إنما يخزن حاصل الطرح بين القيمة المعرفة في الحقل m أو الثابت m وبقية الحقول أو الثوابت (١) و (٢) و ... في الحقل (n) والحقل (o).

مثال ١ : SUBTRACT A FROM B.

قبل التنفيذ A $\boxed{10}$ B $\boxed{15}$
بعد التنفيذ A $\boxed{10}$ B $\boxed{5}$

في المثال (١) فإن فعل الطرح أدى إلى تخزين حاصل عملية التفاضل بين الحقل B والحقل A (حيث تتم عملية طرح A من B) و يلاحظ أن قيمة A لم تتغير بعد تنفيذ عملية الطرح .

مثال ٢ : SUBTRACT A FROM B GIVING C.

قبل التنفيذ A $\boxed{10}$ B $\boxed{15}$ C $\boxed{20}$
بعد التنفيذ A $\boxed{10}$ B $\boxed{15}$ C $\boxed{5}$

يلاحظ في المثال (٢) مع استعمال عبارة GIVING أن قيمة الحقل A قد طرحت من الحقل B ووضعت نتيجة الطرح في المخزن C بغض النظر عن القيمة الموجودة فيه في الأصل . كما أنه يلاحظ عدم تبدل قيمة A و B بعد تنفيذ عملية الطرح . والجداول التالية يوضح أمثلة أوسع عن تعليمة الطرح .

| | SUBTRACT | | | | تعليمة الطرح |
|--------------------------|----------|----|----|----|--------------|
| اسم الحقل : | A | B | C | D | |
| قيمة الحقل قبل التنفيذ : | 5 | 20 | 25 | 30 | |
| قيمة الحقل بعد التنفيذ : | | | | | |
| SUBTRACT A FROM C. | 5 | 20 | 20 | 30 | |
| SUBTRACT A B FROM C. | 5 | 20 | 0 | 30 | |
| SUBTRACT A B FROM C | 5 | 20 | 25 | 0 | |
| GIVING D. | | | | | |
| SUBTRACT 10 FROM C, D. | 5 | 20 | 15 | 20 | |

٢ - ٤ - ٣ تعليمية الضرب MULTIPLY

تأخذ تعليمية الضرب في لغة كويول الشكليات الأماميين العاملين التاليين :

$$1) \quad \underline{\text{MULTIPLY}} \quad \left\{ \begin{array}{l} \text{identifier - 1} \\ \text{literal - 1} \end{array} \right\} \quad \underline{\text{BY}} \quad \text{identifier - 2} \quad \left[\text{identifier - 3} \right]$$

$$2) \quad \underline{\text{MULTIPLY}} \quad \left\{ \begin{array}{l} \text{identifier - 1} \\ \text{literal - 1} \end{array} \right\} \quad \underline{\text{BY}} \quad \left\{ \begin{array}{l} \text{identifier - 2} \\ \text{literal - 2} \end{array} \right\} \quad \underline{\text{GIVING}}$$

identifier - 3 [identifier - 4]

وتعليمية الضرب تقوم بضرب الحقل (١) بالحقل (٢) وتخزين الجداء في الحقل (٣) و (٤). وذلك مع عدم استعمال عبارة GIVING. بينما في حال استخدم عبارة GIVING فإن نتيجة الضرب (جداء الحقل (١) و (٢)) تخزن في الحقل (٣) و (٤).

مثال ١ : MULTIPLY A BY B.

A 5 B 10 قبل التنفيذ :

A 5 B 50 بعد التنفيذ :

و يلاحظ في هذا المثال أن جداء الحقل A و B قد وضع في الحقل B وتبدلت قيمته من 10 الى 50 بينما الحقل A لم يتبدل.

مثال ٢ :

MULTIPLY A BY B GIVING C.

A : 5 B : 10 C : 15 : قبل التنفيذ :

A : 5 B : 10 C : 50 : بعد التنفيذ :

يلاحظ أن قيمة الحقل C قد استبدلت بالقيمة 50 وهي جداء عملية الضرب A في B، وأن قيمة الحقلين A و B بعد التنفيذ لم تتبدل. والجدول التالي يوضح أمثلة أخرى عن تعليمة الضرب.

تعلية الضرب MULTIPLY

| اسم الحقل : | A | B | C |
|------------------------------|----|----|----|
| قيمة الحقل قبل التنفيذ : | 5 | 10 | 20 |
| قيمة الحقل بعد التنفيذ : | | | |
| MULTIPLY B BY A GIVING C. | 5 | 10 | 50 |
| MULTIPLY A BY B GIVING C. | 5 | 10 | 50 |
| MULTIPLY A BY B | 5 | 50 | 20 |
| MULTIPLY B BY A. | 50 | 10 | 20 |
| MULTIPLY A BY 3 GIVING B, C. | 5 | 15 | 15 |

٢ - ٤ - ٤ : تعلية التقسيم DIVIDE

عملية التقسيم تنفذ في لغة كويول بواسطة الفعل DIVIDE وتأخذ تعليمة التقسيم عندئذ الاشكال الأساسية التالية :

$$1) \quad \underline{\text{DIVIDE}} \left\{ \begin{array}{c} \text{identifier - 1} \\ \text{literal - 1} \end{array} \right\} \underline{\text{INTO}} \text{identifier - 2} \left[\text{identifier - 3} \right]$$

$$2) \quad \underline{\text{DIVIDE}} \left\{ \begin{array}{c} \text{identifier - 1} \\ \text{literal - 1} \end{array} \right\} \left\{ \begin{array}{c} \underline{\text{INTO}} \\ \underline{\text{BY}} \end{array} \right\} \left\{ \begin{array}{c} \text{identifier - 2} \\ \text{literal - 2} \end{array} \right\} \quad \underline{\text{GIVING}} \\ \text{identifier - 3} \quad [\text{identifier - 4}]$$

$$3) \quad \underline{\text{DIVIDE}} \left\{ \begin{array}{c} \text{identifier - 1} \\ \text{literal - 1} \end{array} \right\} \left\{ \begin{array}{c} \underline{\text{INTO}} \\ \underline{\text{BY}} \end{array} \right\} \left\{ \begin{array}{c} \text{identifier - 2} \\ \text{literal - 2} \end{array} \right\} \quad \underline{\text{GIVING}} \\ \text{identifier - 3} \quad \underline{\text{REMAINDER}} \quad \text{identifier - 4}$$

ففي الشكل (١) يحفظ حاصل القسمة (قسمة الحقل (٢) على الحقل (١)) في الحقل (٢) وتلغى القيمة الأولية للحقل. وإذا كان هناك أكثر من حقل أي (٣) أو (٤) فإن حاصل قسمة هذه الحقول (المقسومة) تحفظ في نفس مكان القيمة الأولية أي الرقم المقسوم) على التالي.

أما في الشكل (٢) فإن حاصل قسمة الحقل (أو العدد الثابت) (٢) على الحقل (١) باستخدام كلمة INTO أو حاصل قسمة الحقل (١) على الحقل (٢) باستخدام كلمة BY يحفظ في الحقل (٣) بدلاً من القيمة الأولية التي كانت فيه، وإذا كان هناك حقل رقم (٣) فإن حاصل القسمة يحفظ فيه أيضاً.

أما في الشكل (٣) فإن حاصل القسمة يحفظ في الحقل (٣) بينما يحفظ الباقي من عملية التقسيم REMAINDER في الحقل (٤).

مثال ١ :

| | | | | |
|--------|--|----------|---|---------------|
| DIVIDE | | A INTO B | | : |
| A : | <div style="border: 1px solid black; padding: 2px;">20</div> | B : | <div style="border: 1px solid black; padding: 2px;">100</div> | قبل التنفيذ : |
| A : | <div style="border: 1px solid black; padding: 2px;">20</div> | B : | <div style="border: 1px solid black; padding: 2px;">5</div> | بعد التنفيذ : |

مثال ٢ : DIVIDE B BY A GIVING C.

أو DIVIDE A INTO B GIVING C.

قبل التنفيذ : A : 20 B : 100 C : 13

بعد التنفيذ : A : 20 B : 100 C : 5

يلاحظ في المثال (١) أن قيمة الحقل B بعد التقسيم تأخذ حاصل قسمة العدد 100 على 20 وهو الرقم 5. أما في المثال الثاني فإن حاصل قسمة الحقل B على A يوضع في الحقل C (أي 5) وبغض النظر عن القيمة الأولية الموجودة أصلاً فيه وهي (13).

مثال ٣ : DIVIDE A INTO B GIVING C

REMAINDER D.

قبل التنفيذ : A : 20 B : 102 C : 13 D : 15

بعد التنفيذ : A : 20 B : 102 C : 5 D : 2

يلاحظ في هذا المثال أن حاصل قسمة الحقل B على A قد خزنت في الحقل ٢ عوضاً عن القيمة (13) بينما تم حفظ الباقي من عملية التقسيم وهو العدد 2 في الحقل D بدلاً من القيمة 15. ويلاحظ أيضاً أن قيمة الحقلين A و B لم تتبدل بعد تنفيذ عملية التقسيم.

والجدول التالي يوضح أمثلة مختلفة عن تعليمة التقسيم.

| | تعليمة التقسيم DIVIDE | | | |
|----------------------------|--------------------------|----|----|--------------------------|
| Data name : | A | B | C | اسم الحقل : |
| Value before execution : | 10 | 20 | 60 | قيمة الحقل الأولية : |
| Value after execution of : | | | | قيمة الحقل بعد التنفيذ : |

| | | | |
|---------------------------|----|----|----|
| DIVIDE 4 INTO B. | 1φ | 5 | 6φ |
| DIVIDE 4 INTO B GIVING C. | 1φ | 2φ | φ5 |
| DIVIDE B BY 10 GIVING A. | φ2 | 2φ | 6φ |
| DIVIDE A INTO B, C. | 1φ | φ2 | φ6 |
| DIVIDE A INTO B GIVING C. | 1φ | 2φ | φ2 |
| DIVIDE 3 INTO A GIVING B | 1φ | 3 | 1 |
| REMAINDER C. | | | |

٢-٤-٥ تعليمة احسب COMPUTE

تعليمة COMPUTE يمكن أن تنفذ أي عملية حسابية سابقة مثل الجمع، الطرح، الضرب، أو التقسيم بالإضافة إلى إمكانية احتوائها على عمليات مختلفة (أو مركبة) أي جمع مع طرح مع ضرب أو مع تقسيم. فمثلاً التعبير الحسابي التالي :

$$Y = 5 (A + B) / C$$

وفيه قيمة المتغير Y عبارة عن عمليات جمع وضرب وتقسيم، فلأردنا كتابته بالتعليمات السابقة فيكون :

ADD A B GIVING Y.

MULTIPLY 5 BY Y.

DIVIDE C INTO Y.

بينما عند استخدام تعليمة COMPUTE فيمكن كتابة هذه العمليات الحسابية على النحو التالي :

$$COMPUTE Y = 5 * (A + B) / C.$$

و يكون الشكل العام لتعليمة COMPUTE :

$$COMPUTE identifier - 1 = arithmetic expression$$

حيث أن التعبير الرياضي (أو الحسابي) Arithmetic expression يجب أن يصاغ

على النحو التالي :

- ١ - ان الاشارات + ، - ، * ، / تشير الى الجمع ، الطرح ، الضرب ، التقسيم والرفع الى قوة على التوالي .
 - ٢ - يمكن أن يحتوي على أسماء حقول أو ثوابت literal أو اشارات حسابية وأقواس . ويجب أن يكون هناك فراغ قبل الاشارة الحسابية وفراغ بعدها .
 - ٣ - تستخدم الأقواس () لتوضيح تنفيذ العمليات الحسابية المتسلسلة في نفس التعبير الحسابي . ويجب أن يكون أي تعبير حسابي محصور بين قوسين موافق للشروط العامة للتعبير الحسابي في تعليمة COMPUTE . ويسبق القوس الأول فراغ ويجب أن يلي القوس الآخر فراغ أيضاً .
- وتقوم تعليمة COMPUTE بحساب القيمة المينة في التعبير الحسابي على عین إشارة المساواة '=' ثم تخزن نتيجة العمليات الحسابية المختلفة في الحقل (أو المتغير) المين الى يسار إشارة المساواة . ويجري تقييم الصيغة الحسابية على النحو التالي :
- ١ - يتم حساب قيمة الصيغة داخل الأقواس أولاً وبشكل منفصل .
 - ٢ - وضمن الصيغة الحسابية الواحدة يتم تنفيذ العمليات الحسابية من جهة اليسار الى اليمين وفق الترتيب التالي :
- الرفع الى قوة
 - الضرب والتقسيم
 - الجمع والطرح
- ٣ - وعند وجود عمليتين حسابيتين من نفس المستوى ، مثلاً : جمع وطرح يتم تنفيذهما اعتباراً من جهة اليسار .
- وتجدر الاشارة هنا الى أن بعض أنواع الحاسبات لا تسمح باستخدام تعليمة COMPUTE لأنها تعتبرها أقرب الى اللغات العلمية منها الى اللغة التجارية كإبول فيجب الرجوع الى مراجع الشركة الصانعة للحاسب عند تطبيق هذه التعليمة .

والجدول التالي يوضح أمثلة عن تعليمة COMPUTE :

| Data name : | A | B | C | اسم الحقل : |
|----------------------------|---|---|------|---|
| Value before execution : | 4 | 6 | 20 | القيمة الأولية للحقل : |
| Value After execution of : | | | | قيمة الحقل قبل تنفيذ أي من التعليمات : |
| COMPUTE C = A + B. | 4 | 6 | 10 | جمع بسيط |
| COMPUTE C = A + B * 5 | 4 | 6 | 34 | الضرب يتم قبل الجمع |
| COMPUTE C = (A + B) * 5 | 4 | 6 | 50 | حساب الأقواس يتم قبل |
| COMPUTE C = A ** 2. | 4 | 6 | 16 | تعاادل جبرياً $c = a^2$ |
| COMPUTE C = B ** A. | 4 | 6 | 1296 | تعاادل جبرياً $c = b^a$ |

٢- ٥ أسئلة وتمارين

سؤال ١- ما هي الأخطاء التي يتوقع وجودها مترجم لغة كوبيول عند ترجمته لهذا الجزء من البرنامج ؟

DATA DIVISION.

FILE - SECTION.

FD IN - FILE.

LABEL RECORD IS STANDARD.

01 A - REC.

03 A - 1 PIC x (6).

03 A - 2.

05 A - 3 PIC x (8).

07 A - 4 PIC 999.

| | | | |
|----|--------|-----|--------|
| Ø7 | A - 5 | PIC | xxxx. |
| Ø7 | A - 6. | | |
| Ø5 | A - 7 | PIC | x (2Ø) |
| Ø2 | A - 8 | PIC | 9 9. |

سؤال ٢ - إذا كان لدينا ملفان على بطاقات مثقبة الأول اسمه A - FILE والآخر اسمه B - FILE . وهناك ملف ثالث محجوز على القرص المغنط اسمه C - FILE لأجل كتابة السجلات عليه .

هذه الملفات تحتوي السجلات على التالي C - REC , B - REC , A - REC وقسم الاجراءات في برنامج كوبول مصمم لانجاز ما يلي :

أ - فتح الملفات

ب - قراءة الملف A - FILE

ج - قراءة الملف B - FILE وكتابة سجل على الملف C - FILE

د - تخزين (أو حفظ) السجل A - REC في السجل C - REC

هـ - في نهاية الملف A - FILE يجب اعادة فتحه وقراءته من أوله واعادة المعالجة من الخطوة رقم (ب)

و - في نهاية الملف B - FILE تطلق الملفات وينتهي البرنامج .

والمطلوب تدقيق هذا الجزء من برنامج كوبول الذي من المفترض أن يتجزه هذه الاجراءات :

PROCEDURE DIVISION.

OPEN INPUT A - FILE OUTPUT B - FILE.

A - READ.

READ A - FILE AT END GO TO A - END.

READ B - FILE AT END CLOSE B - FILE STOP RUN.

OPEN OUTPUT C - FILE.

WRITE C - FILE FROM B - FILE.

MOVE A - REC TO C - REC.

GO TO A - READ.

A - END.

CLOSE A - FILE.

MOVE A - REC TO C - REC.

OPEN INPUT A - FILE.

GO TO A - READ.

سؤال ٣ - أكتب التعليمات التي تقوم بها يلي :

أ - ملء المخزن REC - COUNTS بالأصفار

ب - إعطاء القيمة 19 للمخزن المسمى A - QTY

ج - إعطاء القيمة 1 للمخزن المسمى B - QTY

د - وضع العبارة COBOL EXERCISE في المخزن المسمى A - AREA

سؤال ٤ - أكمل الجدول التالي للتعليمات الصحيحة فقط، حيث أن قيم الحقول A, B, C, D معطاة كقيمة أولية قبل تنفيذ التعليمات، وبين سبب عدم صحة التعليمات الخطأ

Field - name :

Initial - value :

MOVE C TO SPACE.

MOVE ZERO TO A, B, C.

MOVE SPACE TO A, B.

MOVE D TO B.

MOVE 'X' TO A.

| A | B | C | D |
|---|---|---|---|
| 4 | 5 | 6 | 7 |
| - | - | - | - |
| - | - | - | - |
| - | - | - | - |
| - | - | - | - |

الفصل الثالث

الأقسام الأخرى في برنامج كوبول

٣- ١ ملحة عامة

علمنا من الفصل الأول أن برنامج كوبول يتألف من أربعة أقسام رئيسية وهي قسم التعريف وقسم محيط البرنامج وقسم البيانات وقسم الاجراء وبينا في الفصل الثاني التعليمات المهمة الأولية التي يمكن أن ترد في قسم الاجراء. أما في هذا الفصل فسوف ننظر في كيفية كتابة الأقسام الأخرى وبشكل خاص قسم البيانات DATA DIVISION لأنه القسم الأكثر أهمية بالنسبة لباقي الأقسام. كما سوف يتم في هذا الفصل شرح أرقام المستوى وعبارة الصورة Picture ثم الفصل الخاص بتوصيف البيانات في التخزين الداخلي للبرنامج ثم أشكال اخراج البيانات وإعطاء الصورة المطلوبة. وبنهاية هذا الفصل يمكن للدارس الإلمام بكل المعلومات الضرورية لكتابة برنامج كوبول أولي يكون نواة لبرنامج أكثر تقدماً في الفصول اللاحقة.

٣- ٢ قسم التعريف IDENTIFICATION DIVISION

يعتبر قسم التعريف أول قسم من الأقسام الأربعة الذي يتألف منها برنامج كوبول ويرد في أول البرنامج. ووظيفة هذا القسم تزويد البرنامج بالمعلومات التعريفية عن البرنامج نفسه مثل : اسم البرنامج ومؤلف البرنامج ، وتاريخ البرنامج

(*) Identifying information

وتاريخ كتابته واسم مركز الحاسب وتاريخ ترجمة البرنامج .. الخ. ويتألف القسم هذا من عنوان أساسي له وأسماء فقرات تالية له يمكن أن تصل الى ست فقرات وفق الشكل التالي :

IDENTIFICATION DIVISION.

PROGRAM - ID. program - name.

[AUTHOR. Comment - entry.]

[INSTALLATION. Comment - entry.]

[DATE - WRITTEN. Comment - entry.]

[DATE - COMPILED. Comment - entry.]

[SECURITY. Comment - entry.]

ويلاحظ من الشكل أن هناك فقط فقرة واحدة بعد اسم القسم اجبارية وهي PROGRAM - ID وتمثل اسم البرنامج أما باقي الفقرات فهي اختيارية وتدل على التوالي : اسم المؤلف (المبرمج)، اسم مركز الحاسب، تاريخ كتابة البرنامج، تاريخ ترجمة البرنامج، وضمان سلامة البيانات والبرنامج. ويرد في هذه الفقرات بيانات تعتبر بمثابة تطبيق وتوثيق للبرنامج لا أكثر من أجل سهولة الرجوع اليه في المستقبل ولذلك دعيتم بيانات تطبيقية. Comment - entry.

مثال :

IDENTIFICATION DIVISION.

PROGRAM - ID. PROG #1.

AUTHOR. AUTHORS.

INSTALLATION. IPA COMPUTER CENTER.

DATE - WRITTEN. FEBRAURY 4, 1983.

DATE - COMPILED. FEBRAURY **, 1983.

SECURITY. THERE IS PASSWORD FOR USER.

و يُرْمَزُ هذا المثال وفق قواعد الكتابة على الورقة الخاصة بلغة كويول ويجب أن تبدأ جميع أسماء هذه الفقرات من المنطقة A (أي الأعمدة ٨ - ١١). أما محتوى الفقرة فيمكن أن يكتب في المنطقة B (أي الأعمدة ١٢ وما بعد إلى ٧٢).

٣-٣ قسم البيئة (المحيط) ENVIRONMENT DIVISION

يعتبر قسم محيط البرنامج القسم الثاني في برنامج كويول من حيث تسلسله في البرنامج ووظيفة هذا القسم تتلخص في نقطتين :

- تعريف الحاسب المستخدم في عملية ترجمة البرنامج وتنفيذه (عادة ما يكون الحاسب نفسه هو المستخدم في عملية الترجمة وعملية التنفيذ). و يظهر هذا التعريف في فصل مكونات الحاسب Configuration section.
- ربط الملفات المستخدمة في البرنامج مع وحدات الإدخال والإخراج المتصلة بالحاسب مباشرة. وهذا يظهر في فصل الإدخال والإخراج ضمن فقرة ضبط الملفات INPUT - OUTPUT SECTION, FILE - CONTROL.

لذلك فقد يختلف هذا القسم في برنامج كويول باختلاف الحاسب المستخدم ويُنصح بالرجوع الى مراجع الشركة الصانعة للحاسب لتحديد معطيات هذا القسم. و يأخذ فصل تعريف مكونات الحاسب الشكل العام التالي :

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE - COMPUTER. Computer - name.

OBJECT - COMPUTER. Computer - name.

وتبدأ كتابة أسماء الفقرات في المنطقة A (الأعمدة ٨ - ١١) وبقية المعطيات في المنطقة B (الأعمدة ١٢ - ٧٢) .

أما الفصل الثاني الأكثر أهمية في هذا القسم هو فصل الإدخال والإخراج ويحتوي على فقرة تدعى FILE - CONTROL وضمنها يُعرف كل ملف من الملف المستخدمة في البرنامج بعبارة خاصة تدعى SELECT مع كلمة ASSIGN وتعني اختيار وحدة الإدخال والإخراج وتميئها لاسم الملف المنطقي المستخدم ضمن البرنامج وتقوم هذه العبارة بالربط بين الوحدة المخصصة والملف المستخدم لأجل تعريف البرنامج بجهة مصدر أو مستقبل البيانات من وإلى البرنامج . وتخصيص أو تعيين وحدات الإدخال والإخراج يختلف من حاسب لآخر لذلك ينصح بالعودة الى مراجع الشركة الصانعة للحاسب لهذا الغرض .

والشكل التالي يوضح كيفية هذا الفصل لحاسب شركة IBM :

INPUT - OUTPUT SECTION.

FILE - CONTROL.

SELECT CARD - FILE ASSIGN TO UT - S - SYSIN.

SELECT PRINT - FILE ASSIGN TO UT - S - SYSOUT.

و يقصد بعبارتي SELECT ما يلي : تخصيص وحدة الادخال المسماة UT - S - SYSIN لملف الادخال المسمى في البرنامج باسم CARD - FILE وكذلك تخصيص وحدة الاخراج المسماة في نظام التشغيل UT - S - SYSOUT لملف الاخراج المسمى في البرنامج باسم PRINT - FILE .

(٥) حسب توصيف IBM 3033 ونظام التشغيل OS/MVS .

أما إذا أردنا كتابة نفس الفصل لحاسب TI 990 فإنه يأخذ الشكل التالي :

INPUT - OUTPUT SELECT.

FILE - CONTROL.

SELECT CARD - ASSIGN TO INPUT "SYN"

SELECT PRINT - FILE ASSIGN TO PRINT "SYN"

وكما سبق فإن أسماء الفقرات تكتب ابتداء من المنطقة A (الأعمدة ٨ - ١١) وعبارة SELECT تبدأ من المنطقة B (الأعمدة ١٢ - ٧٢).

٣ - ٤ قسم البيانات DATA DIVISION

يأتي قسم البيانات في برنامج كوبرول بعد قسم محيط البرنامج ويصف هذا القسم جميع الحقول وأماكن التخزين في الذاكرة اللازمة للبرنامج. كما يحتوي في وصفه على حجم كل حقل بالرمز Characters وكذلك نوعية هذا الحقل مثل رقمي أو أبجدي أو غير رقمي. ويوضح أيضا هيكل أو تركيب البيانات المستخدمة. والفقرة التالية تبين كيفية وصف هذه المعطيات بواسطة عبارة الصورة Picture وأرقام المستويات المتدرجة.

٣ - ٤ - ١ عبارة PICTURE

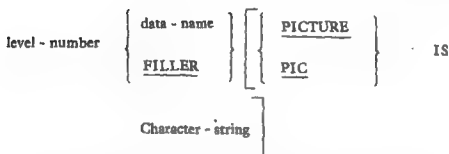
في قسم البيانات يجب وصف البيانات وتحديد أسمائها وكذلك حجم البيانات ونوعيتها ويتم ذلك بواسطة عبارة الصورة PICTURE. ومبدئيا هناك ثلاثة أنواع مهمة من البيانات وهي : الأبجدية ويرمز لها بالحرف A والرقمية بالرقم 9. والحرفية - الرقمية ٥٥ بالرمز X. أما حجم البيانات (أي طول الحقل المستخدم، فيحسب بعدد

(٥) حسب توصيف TI 990 نظام التشغيل 12 x D وكلمة SYN يقصد بها اسم الملف الرمزي والذي يجب أن يعرف للحاسب (نظام التشغيل) بأمر COMMAND يدعى AS (ASSIGN SYNONYM).

(٥٥) Alphanumeric

الرموز (أحيانا يكون كل رمز مساوياً لبايت Byte واحدة). و يعبر عن ذلك بتكرار رمز نوع البيانات بعدد المرات مساوياً لطول الحقل المرغوب. فمثلاً إذا كان الحقل أبجدياً ومساوياً لأربع رموز فيكتب على شكل AAAA أو (4) A، أما إذا كان الحقل رقمياً وطوله ثلاثة أرقام فيكتب وصفه على الشكل 999 أو (3) 9 أما إذا كان الحقل غير رقمي وكان طوله فرضاً ستة رموز فيكتب (6) X أو XXXXXX.

وتأخذ هذه العبارة الشكل العام التالي الآتي :



حيث يبين رقم المستوى level - number تسلسل الحقل الموصوف و data - name اسم المتغير أو الحقل الموصوف و character - string يمثل نوع وحجم الحقل الموصوف. أما الكلمات PICTURE, PIC, IS فهي كلمات كويول محجوزة ويقصد بكلمة FILLER اسم لحقل لا يستخدم من قبل المبرمج ويمثل حقل DUMMY ضمن البيانات.

٣- ٤- ٢ أرقام المستويات LEVEL NUMBERS

تصنف حقول البيانات data - items في لغة كويول الى نوعين : حقول اجمالية group وحقول ابتدائية (أولية) elementary. والحقل الاجمالي عبارة عن مجموعة حقول ابتدائية أما الحقل الابتدائي فهو الحقل الذي لا يمكن تجزئته الى حقول أخرى. والشكل التالي يبين مثالا على هذا التصنيف :

| STD - CARD | | | | | |
|------------|---------------------|-----------------------|----------|-------------------------------|---------------------|
| STD - NO | STD - NAME | | MARKS | | SPACE (NOT USED) |
| | STD - LAST- NAME | STD - FIRST - NAME | SCORE | GRADE | |
| 1 78 | 22 23 | 37 38 | 41 42 43 | 80 | |
| numeric | Alphabetic | | numeric | Alpha - غير مستخدم numeric | |

ففي هذا الشكل يتضح هيكل بيانات لسجل (حقل اجمالي) يسمى
STD - CARD . ويجزأ الى حقول فرعية :

STD - NO , STD - NAME , MARKS

والحقل الجزئي STD - NAME مقسم الى حقلي آخرين

STD - LAST - NAME , STD - FIRST - NAME

وكذلك الحقل MARKS مقسم الى حقلي :

SCORE , GRADE

فالحقول الاجالية في هذا الشكل هي STD - CARD

و STD - NAME , MARKS بينما الحقول الابتدائية هي :

STD - NO , STD - LAST - NAME , STD - FIRST - NAME , SCORE , GRADE.

ولاجل توصيف هذا التركيب من البيانات في لغة كويول تستخدم أرقام المستوى
للدلالة على تدرج توزيع الحقول من اجمالي الى جزئي ، والحقل الاجمالي الأول يُعطى
الرقم 01 ويُعطى الحقل أو الحقول المكونة له أرقاماً أكبر مثل 02 وهكذا لغاية الرقم 49
ضمناً . (كما يوجد هناك الأرقام 66 و 77 و 88 وسوف يرد ذكرها وكيفية استخدامها

في الفصول التالية). ويمكن وضع الشكل السابق من البيانات على صيغة سجل في لغة كويول على النحو التالي :

φ1 STD - CARD.

φ5 STD - NO PICTURE IS 9 (7).

05 STD - NAME.

φ7 STD - LAST - NAME PICTURE IS A (15).

φ7 STD - FIRST - NAME PICTURE IS A(15).

φ5 MARKS.

1φ SCORE PICTURE IS 9 (4).

1φ GRADE PICTURE IS X.

φ5 FILLER PICTURE IS X (38).

يلاحظ من هذا التوصيف أن الحقول الابتدائية هي فقط التي تشتمل على عبارة PICTURE كما يلاحظ أيضا أن أرقام المستوى ليس بالضرورة أن تكون متسلسلة . أما الحقل الأخير المسمى FILLER فهو يشير الى اسم حقل لجزء من السجل غير مستخدم (أي مغفل في البرنامج) ويجب دائما وصفه بالنوع X لأنه من الجائز أن يكون محتويا على رموز حرفية ورقمية أو يكون فارغاً . هذا ويجوز استخدام كلمة PIC للدلالة على PICTURE وكلمة IS ليست اجبارية عند كتابة هذه العبارة كما يجب الاشارة هنا إلى أن أرقام المستوى نكتب ابتداء من الأعمدة ٨ - ١١ بالنسبة للحقل الأول الاجمالي فقط أما بقية الأرقام فيمكن أن ترد ضمن هذه المنطقة أو بعدها كما أن وجود أكثر من فراغ بين الكلمات في التوصيف ليس خطأ بل يسهل عملية القراءة والمراجعة .

٣-٤-٣ فصل توصيف الملفات FILE SECTION

يعتبر فصل توصيف الملفات الفصل الأول في قسم البيانات في لغة كوبول وهذا الفصل يشتمل على توصيف لكل ملف تم ذكره في عبارة SELECT في قسم محيط البرنامج وإذا لم يكن هناك ملفات ضرورية في البرنامج المكتوب فلا داعي لذكر هذا الفصل لذلك يمكن اعتباره فصلاً اختيارياً.

وفصل توصيف الملف يحتوي عادة على وصف للملف يليه توصيف للسجل الخاص بهذا الملف. وقد وصفنا في الصفحات القليلة السابقة سجل الملف طالب STD-CARD ولنفرض أنه من ملف اسمه STD-FILE فلأجل وصف الملف نستخدم الشكل العام الأولي لوصف الملف ويدعى عبارة FD* ويكتب على النحو التالي :

FD file - name

$$\text{LABEL} \left\{ \begin{array}{cc} \text{RECORD} & \text{IS} \\ \text{RECORDS} & \text{ARE} \end{array} \right\} \left\{ \begin{array}{c} \text{OMITTED} \\ \text{STANDARD} \end{array} \right\}$$

[RECORD CONTAINS integer - 1 CHARACTERS]

[DATA RECORD IS data - name - 1]

وعبارة LABEL تصف خلاصة وصف الملف المسجل أو الذي سوف يسجل على وحدات التخزين المغناطيسي في حال كونها STANDARD أما كلمة OMITTED فتستخدم مع ملف الإدخال من بطاقات مثقبة أو ملف الإخراج على طابعة أو مع ملف الشريط المغناطيسي أحياناً عندما لا يحتوي على LABEL خاصة به.

(*) تعني File Description

والعبارتان التاليتان الاختياريتان تصفان حجم السجل في هذا الملف واسم
السجل الخاص به على التوالي .
وعندئذ يمكن كتابة وصف الملف FILE - STD على النحو التالي :

FD STD - FILE

LABEL RECORD IS OMITTED •

RECORD CONTAINS 80 CHARACTERS

DATA RECORD IS STD - CARD.

وحسب قواعد لغة كوبول ترد كلمة FD ابتداء من الأعمدة ٨ - ١١ أما بقية
العبارات في هذه الجملة فتبدأ بعد العمود ١٢ إلى ٧٢.

٣ - ٤ - ٤ فصل التخزين الداخلي WORKING - STORAGE SECTION

يستخدم هذا الفصل في قسم البيانات لأجل تخزين النتائج الوسيطة من المعالجة
وكذلك تعريف وتوصيف المعطيات الثابتة في البرنامج أو بمعنى آخر يحتوي على
توصيف للبيانات التي يحتاجها البرنامج بدون أية قراءة من خارج البرنامج .

وهذا الفصل يحتوي على نوعين للبيانات الموصوفة :

- النوع الأول الحقول (المتغيرات) المستقلة والتي تكون أولية بحد ذاتها ولا تحتاج
إلى تمهئة. وهذه الحقول تأخذ رقم المستوى 77 وعادة تذكر في هذا الفصل قبل النوع
الآخر ••، كما ويمكن أن ترد بعده.

(•) يفترض أنه موجود على بطاقات مثقبة .

(••) إن الانتهاء الحديث في البرمجة يقضي بتجميع الحقول المستقلة التي ليس لها علاقة تركيبة مع غيرها في مجموعة (اجالية
من الحقول ثم تأخذ رقم المستوى 01 وبقيّة الأرقام المتدرجة .

- النوع الثاني الحقول (المتغيرات) الاجمالية والتي يمكن تخزينها الى حقول جزئية وتأخذ هذه الحقول رقم المستوى ١٠ والحقول الجزئية تأخذ أرقاماً متدرجة أخرى كما سلف ذكره في الفقرات السابقة .

والشكل التالي يوضح كيفية توصيف البيانات في هذا الفصل :

WORKING - STORAGE SECTION.

| | | |
|----|--------------|---------------------------|
| 77 | TOTAL - AREA | PIC 9 (4) VALUE ZEROS. |
| 77 | MIN - MARK | PIC 99 VALUE 6٠. |
| ٠1 | HDR - LINE. | |
| ٠5 | FILLER | PIC X (1٠) VALUE SPACES. |
| ٠5 | FILLER | PIC X (6) VALUE 'I.P.A.'. |
| ٠5 | FILLER | PIC X (1٠) VALUE SPACES. |

٣ - ٤ - ٥ عبارة تحديد القيمة VALUE

لقد ورد في الفقرة السابقة كلمة VALUE بجانب توصيف الحقل وتعني هذه العبارة تحديد قيمة ثابتة في الحقل أو اعطائه قيمة أولية لا تتبدل أثناء تنفيذ البرنامج إلا بعملية نقل قيمة أخرى على نفس الحقل بواسطة تعليمة MOVE .

وتأخذ عبارة تحديد القيمة الشكل العام التالي :

| | | | |
|----------------|-------------|---|-----------------------|
| level - number | data - name | <div style="display: inline-block; vertical-align: middle;"> <div style="border: 1px solid black; padding: 5px; text-align: center;">PICTURE</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">PIC</div> </div> | IS Character - string |
|----------------|-------------|---|-----------------------|

[VALUE IS literal]

حيث أن كلمة literal تبقى قيمة محددة ويمكن تصنيفها الى ثلاثة أنواع :

١ - ثابت رقمي مثل القيمة 60 في الحقل MIN - MARK

٢ - ثابت غير رقمي (رقمي / حرفي) و يوضع بين اشارتي ` مثل 'I.P.A.' في الحقل

المسمى FILLER

٣ - ثابت شكلي (figurative) وهو عبارة عن كلمات محجوزة في لغة كوبرول لترمز الى ثوابت شائعة الاستخدام ومقروءة بشكل أفضل . مثل كلمة ZERO (وللجمع تستخدم ZEROS في الحقل المسمى TOTAL - AREA وكلمة SPACE (للجمع تستخدم SPACES) كما في الحقل المسمى FILLER .

وتجدر الإشارة هنا بأنه لا يجوز تحديد قيمة غير رقمية في حقل موصوف بالرمز 9 (رقمي) مثل :

φ5 A PIC 9 VALUE Y. (غير صحيح)

كما يجب دائما عند تحديد القيمة غير الرقمية في الحقل غير الرقمي وضع اشارتي ` ` (الفاصلة العلوية) لتحديد بداية ونهاية القيمة الثابتة . مثل :

φ5 B PIC X VALUE 8. (غير صحيح)

وكأمثلة صحيحة نذكر :

77 C PIC 9 VALUE 9.

φ1 D PIC XXX VALUE 'IPA'.

(هـ) الكلمات المفردة والجمع متعادلة في لغة كوبرول أي SPACE تعادل ولما نفس الوظيفة مثل SPACES بل ذكرت بالجمع لأجل القراءة الواضحة باللغة الانكليزية .

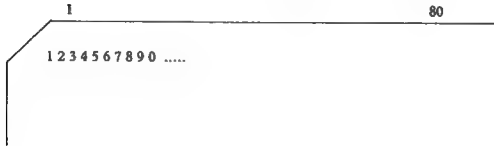
٣- ٤- ٦ الفاصلة العشرية المفترضة ASSUMED DECIMAL POINT

من أجل توفير المساحة في وساطات ادخال البيانات (مثل البطاقة المثقبة سابقاً أو الوسائط المغنطيسية) لا يسمح بادخال الفاصلة العشرية ضمن رموز الرقم أو الحقل المراد قراءته الى البرنامج وإنما يفترض وجود هذه الفاصلة العشرية في الرقم، و يستطيع الحاسب (من خلال البرنامج) تمييز الجزء العشري والجزء الصحيح من الرقم عن طريق وصف صورة الحقل PICTURE التي تحتوي إشارة معينة وهي الرمز V للدلالة على مكان وجود هذه الفاصلة العشرية في الرقم .. فمثلاً الرقم 12345 يقرأ على أنه 123.45 إذا كان موصوفاً في قسم البيانات بعارة PICTURE على النحو التالي :

05 AMOUNT PIC IS 999V99.

ولا يأتي الرمز في عبارة «الصورة» إلا مع النوع الرقمي من البيانات وهو الرمز 9. ويجب التأكيد هنا أن الرمز V يعني وجود فاصلة عشرية في الرقم الموصوف ولكنه لا يُعد أو يحسب ضمن حجم الرقم أو الحقل. فمثلاً الحقل المذكور أعلاه AMOUNT يكون حجمه خمسة أماكن (بايت) فقط ضمن الذاكرة.

ولزيادة الإيضاح : إذا كان لدينا بطاقة مثقبة تحتوي على الأرقام التالية والمتصلة مع بعضها :



وكانت هذه البطاقة موصوفة في قسم البيانات على النحو التالي :

Ø1 PUNCH - CARD.

| | | | | |
|----|---------|-----|--------|---|
| Ø5 | FLD - A | PIC | 9V99. | : |
| Ø5 | FLD - B | PIC | 99V9. | |
| Ø5 | FLD - C | PIC | 9. | |
| Ø5 | FLD - D | P2C | V999. | |
| Ø5 | FILLER | PIC | X(7Ø). | |

FLD - A, FLD - B, FLD - C, FLD - D : فتكون قيمة الحقول :

كما يلي على التوالي :

1.23 , 45.6 , 7 , 890

٣ - ٤ - ٧ اظهار البيانات الرقمية EDITING NUMERIC DATA

عادة لا تحتوي البيانات الرقمية المدخلة سوى أرقام فقط (digits) ولكن بالمقابل لا بد من قراءة البيانات المخرجة والتي يجب أن تشمل على الرموز المساعدة في فهم الأرقام ومعناها. لذلك تتم عملية ادخال الفاصلة العشرية والفاصلة العادية ورمز العملة على الحقل الرقمي عند إخرجه على الطابعة و يسمى الحقل الذي يظهر الرقم بهذا الشكل المقروء بحقل الاظهار (Edited Field).

وهذه العملية توفر مساحة في الذاكرة لعدم حجز أماكن لهذه الرموز وتعطي امكانية قراءة هذه الأرقام بعد اظهارها وتقديمها للمستفيد من البرنامج.

ولنأخذ في هذا القسم فقط ثلاثة رموز من رموز الاظهار وهي كافية في المراحل الأولية من لغة كوبول. وهذه الرموز الفاصلة العشرية، والفاصلة العادية ورمز العملة.

ولنبدأ بالفاصلة العشرية : فإذا كان لدينا الحقل الرقمي المسمى A - FLD والموصوف على النحو التالي مع حقل الاظهار والخاص به المسمى A - FLD - EDT :

5 PIC 9V99. FLD - A

5 PIC 9.99. EDT - FLD - A

و يلاحظ وجود فاصلة عشرية (Decimal Point) [في نفس مكان الرمز V الذي يدل على وجود الفاصلة في الرقم ضمناً] في حقل الاظهار وهذه الفاصلة العشرية تحسب من حجم الحقل أي أنه يتكون من أربعة أماكن (بايت) يتم حجزها على وسيط الاظهار (الطابعة أو الشاشة) وباستخدام تعليمة النقل MOVE يكون (بفرض قيمة الحقل 567).

FLD - A

| | | |
|---|---|---|
| 5 | 6 | 7 |
|---|---|---|

EDT - FLD - A

| | | | |
|---|---|---|---|
| ? | . | ? | ? |
|---|---|---|---|

قبل تنفيذ النقل :

| | | |
|---|---|---|
| 5 | 6 | 7 |
|---|---|---|

| | | | |
|---|---|---|---|
| 5 | . | 6 | 7 |
|---|---|---|---|

بعد تنفيذ النقل :

MOVE

أما الفاصلة العادية : التي تُحسرين الأرقام للدلالة على قيمة الألوف في الحقل فإنها أيضاً توصف ضمن حقل الاظهار وتحسب عند حساب أماكن (مواضع ، بايت) التي يحجزها الحقل في الذاكرة أو على الطابعة.

وكمثال على ذلك نأخذ الحقل F - FLD الذي نريد إظهاره وفق الحقل EDT - FLD - F والموصوفين كما يلي في قسم البيانات :

5 PIC 9 (5). FLD - F

5 PIC 99,999. EDT - FLD - F

وباستخدام تعليمة النقل MOVE يكون (على فرض أن قيمة الحقل 12345) ما يلي :

| FLD - F | EDT - FLD - F | |
|-----------|---------------|-----------------|
| 1 2 3 4 5 | ? ? , ? ? ? | قبل تنفيذ النقل |
| 1 2 3 4 5 | 1 2 , 3 4 5 | بعد تنفيذ النقل |

أما إشارة العملة المستعملة * : (مثلا الدولار \$ وهي الموجودة في أغلب نسخ لغة كوبرول) فهي تُوصف في حقل الاظهار وعلى الجهة اليسرى منه. كما يمكن أن تجتمع رموز الاظهار معاً في توصيف الحقل الرقمي المراد اخراجه ولنأخذ مثلاً الحقل EDT - FLD - M الذي يمثل حقل اظهار للحقل FLD - M والموصوفين في قسم البيانات على النحو التالي :

¢5 FLD - M PIC 9 (4) V99.
¢5 EDT - FLD - M PIC \$ 9,999.99.

وباستخدام تعليمة النقل MOVE يكون (بفرض القيمة تساوي 123456)

| FLD - M | EDT - FLD - M | قبل تنفيذ النقل : |
|-------------|--------------------|-------------------|
| 1 2 3 4 5 6 | \$? , ? ? ? . ? ? | |
| | | بعد تنفيذ النقل : |
| 1 2 3 4 5 6 | \$ 1 , 2 3 4 . 5 6 | |

(هـ) يمكن استخدام رموز عملة أخرى على شرط أن تكون متوفرة على وسيط الاظهار (الطابعة أو الشاشة).

٣-٥ برنامج كوبول متكامل :

يوجد في الصفحات التالية صورة عن برنامج كوبول متكامل ويقوم هذا البرنامج بقراءة بيانات تحتوي على الاسم والراتب ومكان الإقامة والمعمثرم طباعة البيانات للأشخاص والذين تتحقق فيهم الشروط التالية :

الراتب : أن يكون بين ٢٠٠٠ و ٣٠٠٠ ريال

مكان : الإقامة أن يكون مدينة الرياض RIYADH

العمر : أن يكون أكبر من ٣٦ عاماً.

* برنامج كوبول متكامل

* =====

*

*

* قسم التعريف بالبرنامج :

IDENTIFICATION DIVISION.

PROGRAM-ID. TXT23.

AUTHOR. KHAYAT-JNAID.

*

* يرى هذا التمرين الخطوات الاساسيه في كتابه

* برنامج كوبول

*

* قسم البيئات :

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER. IBM-370.

OBJECT-COMPUTER. IBM-370.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT IN-FILE ASSIGN TO UT-S-INF.

SELECT P-FILE ASSIGN TO UT-S-PRFL.

*

* قسم توصيف البيانات :

DATA DIVISION.

FILE SECTION.

* : توصيف ملف الإدخال :

```
*
FD      IN-FILE      LABEL RECORD IS STANDARD
                        DATA RECORD IS IN-REC.
```

```
01      IN-REC.
03      FILLER      PIC X(05).
03      NAME        PIC A(15).
03      FILLER      PIC X(04).
03      SALARY      PIC 9(04).
03      FILLER      PIC X(11).
03      LOCATION    PIC X(11).
03      AGE         PIC 9(2).
03      FILLER      PIC X(28).
```

* : توصيف ملف الطباعة :

```
*
FD      P-FILE      LABEL RECORD IS OMITTED
                        DATA RECORD IS P-REC.
```

*
يوصف غالباً سجل بطول ١٣٣ حرفاً وهو عرض الطباعة
وتحرك السجلات الأخرى الموضحة في WORKING-STORAGE
إليه وتتم الكتابة باسم هذا السجل P-REC وما يمتدوى
من بيانات مقلدة .

```
*
01      P-REC      PIC X(132).
*
WORKING-STORAGE SECTION.
```

```
*-----*
* توصيف العناوين كما هو واضح في ورقة *
* التوصيف المرفقة PRINT CHART وتسمى *
* العناوين الثابتة أو الثوابت FILLER *
*-----*
```

```
01      HEADER-1.
03      FILLER      PIC X(05) VALUE SPACES.
03      FILLER      PIC X(15) VALUE
```

```

      ' N A M E '
03  FILLER PIC X(4) VALUE SPACES.
03  FILLER PIC X(15) VALUE
      ' S A L A R Y '
03  FILLER PIC X(11) VALUE SPACES.
03  FILLER PIC X(16) VALUE
      ' L O C A T I O N '
03  FILLER PIC XX VALUE SPACES.
03  FILLER PIC X(08) VALUE
      ' A G E '
03  FILLER PIC X(50) VALUE SPACES.

```

```

*-----*
* السطر التالي هو لتوضيف فطر جسم التقرير *
* BODY LINE في السطر الذي يحتوى على *
* البيانات المتغيرة والذي سيتكرر على طول *
* الصفحة *
*-----*

```

```

01  DETIL-1.
03  FILLER PIC X(7) VALUE SPACES.
03  NAME-P PIC A(15).
03  FILLER PIC X(05) VALUE SPACES.
03  SAL-P PIC 9(04).
03  FILLER PIC X(22) VALUE SPACES.
03  LOC-P PIC X(11).
03  FILLER PIC X(07) VALUE SPACES.
03  AGE-P PIC 99.
03  FILLER PIC X(52) VALUE SPACES.

```

```

*
* عدد يستخدم لتجميع
* عدد الاسطر
77  L-CONT PIC 99 VALUE 0.
*
* المقول التاليه تستخدم للتسطير بين
* سطور التقرير على عرض الصفحة
77  L-H PIC X(80) VALUE ALL '='.
77  L-D PIC X(80) VALUE ALL '-'.

```

* : قسم الاجراءات
 *

PROCEDURE DIVISION.

*
 * الخطوه الاولى هي فتح الملفات
 OPEN INPUT IN-FILE
 OUTPUT P-FILE.

*
 * طباعة العناوين في الذكره WT-2
 *
 PERFORM WT-2 THRU EXIT-2.

*
 RED-1.
 READ IN-FILE AT END
 GO TO END-JOB.
 ASK-1.
 IF SALARY > 3000 OR SALARY < 2000
 GO TO RED-1.

*
 ASK-2.
 IF LOCATION NOT = 'RIYADH'
 GO TO RED-1.

*
 ASK-3.
 IF AGE NOT < 36 GO TO RED-1.

*
 * تمريك الحقول من ملف الإدخال IN-FILE
 * المقروء الى حقول سجل الطباعة DETIL-1
 * الموضوعة في قسم توصيف البيانات الإدخليه
 WT-1.

| | | | |
|------|----------|----|---------|
| MOVE | NAME | TO | NAME-P. |
| MOVE | SALARY | TO | SAL-P. |
| MOVE | LOCATION | TO | LOC-P. |
| MOVE | AGE | TO | AGE-P. |


```

*
*
*   كتابة السجل على ملف الطباعة
*
      WRITE  P-REC      FROM  DETIL-1
                                AFTER ADVANCING 3 LINES.
      WRITE  P-REC      FROM  L-D.

*
*
*   اضافة { لعدد الاسطر ثم مقارنته
*   محتواه مع العدد } لان عدد الاسطر
*   المسموح طباعتها في الصفحة الواحدة
*   هو ١٠ , فاذا كان اكبر تطلب الصفحة
*   وتطبع العناوين على الصفحة الجديدة
*   ثم يذهب التنفيذ الى قراءة سجل جديد ,
*   وإذا لم يكن اكبر يذهب التنفيذ الى
*   معالجة سجل جديد مباشرة .
*
      ADD      1          TO      L-CONT.

*
      IF      L-CONT      >      10
      PERFORM  WT-2 THRU EXIT-2.

*
      GO TO   RED-1.

*
*
*   ذكره قلب الصفحة وكتابه العناوين :
*   لاحظ تمريك القيمة صفر لعدد الاسطر
*   WT-2.
      MOVE    ZERO        TO      L-CONT.
      MOVE    SPACES      TO      P-REC.
      WRITE   P-REC        AFTER   PAGE.
      WRITE   P-REC FROM L-H  AFTER 1 LINE.
      WRITE   P-REC FROM HEADER-1 AFTER 1.
      WRITE   P-REC FROM L-H  AFTER 1 LINE.
      EXIT-2.
      EXIT.

*
*   ذكره اغلاق الملفات ونهاية العمل :
*
      END-JOB.
      CLOSE   IN-FILE P-FILE.
      STOP    RUN.

```

شكل مدخلات البرنامج

| | | | | |
|-----------------|------|--------|----|----------|
| ZUHAIK KHAYAT | 2500 | RIYADH | 27 | 00000100 |
| ALI ALGHARDI | 2000 | RIYADH | 29 | 00000200 |
| RIAD MAHMOOD | 5000 | RIYADH | 20 | 00000300 |
| SALEH MONAJED | 3000 | RIYADH | 24 | 00000400 |
| ABDULLAH CHANEH | 2500 | RIYADH | 27 | 00000500 |
| ALI SALEH | 3000 | RIYADH | 28 | 00000600 |
| MOMEN MAHJOUB | 3000 | JEDDAH | 29 | 00000700 |
| IBRAHIM ALMOSA | 2500 | RIYADH | 43 | 00000800 |
| FEAS AHMAD | 2000 | RIYADH | 23 | 00000900 |
| KHALDOON JNAID | 2000 | RIYADH | 33 | 00001000 |
| SALEM ABULLA | 2500 | RIYADH | 32 | 00001100 |
| IBRAHIM ALHARBI | 2500 | RIYADH | 30 | 00001200 |
| IBRAHIM ADTARBI | 2400 | RIYADH | 33 | 00001300 |
| SABER ALYUSEF | 2400 | RIYADH | 33 | 00001400 |
| HAYTHAM ALZOUNH | 3000 | RIYADH | 33 | 00001500 |
| ISA ALMADANI | 3000 | RIYADH | 33 | 00001600 |
| KHALED NJAIM | 3000 | RIYADH | 33 | 00001700 |

٣-٦ أسئلة ترمينات

سؤال ١ - بين الخطأ والصحيح من التعليمات التالية المكتوبة بلغة كوبرول علماً بأن اسم الملف يتبعه كلمة FILE واسم السجل يتبعه كلمة RECORD :

1. OPEN FILE - ONE, FILE - TWO.
2. OPEN INPUT FILE - ONE, FILE - TWO.
3. OPEN INPUT FILE - ONE, FILE - TWO.
4. OPEN FILE - ONE.
5. OPEN EXTEND FILE - ONE, FILE - TWO.
6. OPEN 1 - 0 FILE - ONE, FILE - TWO.
7. READ FILE - ONE.
8. READ FILE - ONE, AT END GO TO E - OF - DATA.
9. READ RECORD - ONE AT END MOVE "NO" TO MORE - DATA.
10. READ FILE - ONE AT END CLOSE FILE - ONE, FILE - TWO
STOP RUN.
11. WRITE RECORD - ONE.
12. WRITE FILE - TWO AFTER 1 LINE.
13. WRITE RECORD - ONE AFTER LINE - NO LINE.
14. WRITE RECORD - TWO AFTER PAGE.
15. CLOSE INPUT FILE - ONE, OUTPUT FILE - TWO.
16. ADD X - FLD 18 Y - FLD GIVING R - FLD
17. ADD X - FLD Y - FLD Z - FLD GIVING R - FLD.
18. ADD A - FLD TO B - FLD.
19. DIVIDE A - FLD INTO 100.
20. DIVIDE 100.25 INTO 4 GIVING R - FLD.

سؤال ٢ - لدينا الجزء التالي من برنامج كويول

PROCEDURE DIVISION.

MAIN - PARA.

PERFORM A.
PERFORM B 3 TIMES.
PERFORM C THRU E.
PERFORM F UNTIL N = 5.
STOP RUN.

A. .

.
.

B. .

.
.

C. .

.
.

D. .

.
.

E. .

.
.

F. .

.
.

ADD 1 TO N.

والمطلوب بيان عدد مرات تنفيذ كل فقرة من الفقرات بعد تنفيذ التعليمات
الأربعة من PERFORM الواردة في الجزء الرئيسي من البرنامج .

سؤال ٣ -

يوجد في الجدول التالي حقلان : الأول يعتبر المصدر SOURCE والثاني المستلم RECEIVING وقد تم وصف كل من هذين الحقليين وبيان قيمة الحقل المصدر. والمطلوب تحديد القيمة الناتجة حسب الوصف المبين للحقل المستلم ؟

| SOURCE FIELD | | RECEIVING FIELD | |
|--------------|--------|-----------------|---------------|
| PICTURE | VALUE | PICTURE | EDITED RESULT |
| 9 (6) | 345678 | 9 (4).99 | |
| 9 (4)V99 | 345678 | 9 (6) | |
| 9 (4)V99 | 345678 | 9 (6).99 | |
| 9 (4)V99 | 345678 | \$ 9 (4) .99 | |

سؤال ٤ - لدينا البرنامج التالي والمكتوب بلغة كوبرول من أجل طباعة جدول بالأشخاص الذين أعمارهم أقل من ٣٠ عاماً وذلك من ملف الموظفين المخزن على بطاقات مثقبة. والمطلوب قراءة هذا البرنامج ودراسته ثم اتمام الجمل أو الكلمات الناقصة واللازمة ليؤدي البرنامج العمل المرسوم له.

000100 IDENTIFICATION DIVISION.

000200 PROGRAM - ID. CPROGI.

000210 AJND. 1

000220 ** THIS PROGRAM WILL PROCESS A FILE OF EMPLOYEE

000230 ** RECORDS AND PRINT THE NAMES UNDER 30 YEARS.

000300 ENVIRONMENT DIVISION.

000400 CONFIGURATION SECTION.

000500 SOURCE - COMPUTER. IBM - 3033.

000600 OBJECT - COMPUTER. IBM - 3033.

000700 INPUT - OUTPUT SECTION.

000800 FILE - CONTROL.

000810 SELECT CARD - FILE ASSIGN TO UT - S - SYSIN.

000900 PRINT - FILE ASSIGN TO UT - S - PRF.

001000

001100 FILE SECTION.

001200 FD PRINT - FILE IS OMITTED

001300 RECORD CONTAINS 132 CHARACTERS.

001310 DATA RECORD IS PRINT - LINE.

001311 01 PRINT - LINE.

001312 05 FILLER PIC X (1).

001313 05 PRINT - NAME

001314 05 FILLER PIC X (2).

001315 05 PRINT - AGE PIC X (3).

001316 05 FILLER PIC X (3).

001317 05 PRINT - SALARY PIC 9 (5).

001318 05 FILLER PIC X (94).

001320 FD CARD - FILE LABEL RECORD IS OMITTED

001321 RECORD CONTAINS 80 CHARACTERS

001322 DATA RECORD IS EMPLOYEE - CARD.

001330 01 EMPLOYEE - CARD.

001340 05 CARD - NAME PIC X (25).

001350 05 CARD - TITLE PIC X (10).

| | | | |
|---------|----|---|---|
| 001360 | 05 | CARD - AGE | PIC 99. |
| 001370 | 05 | FILLER | PIC XX. |
| 001371 | 05 | CARD - SALARY | PIC 9 (5). |
| 001380 | 05 | FILLER | PIC X (36). |
| 001390 | * | <div style="border: 1px solid black; width: 150px; height: 20px; display: inline-block;"></div> 6 | |
| 001400 | | | |
| 001410 | 77 | MORE - INPUT | VALUE 'YES' PIC XXX. |
| 001740 | * | | |
| 001900 | | PROCEDURE DIVISION. | |
| 002000 | | MAIN - ROUTINE. | |
| 002001 | * | HOUSEKEEPING ROUTINE 7 | |
| 002010 | | <div style="border: 1px solid black; width: 50px; height: 15px; display: inline-block;"></div>INPUT CARD - FILE | |
| 0022100 | | OUTPUT PRINT - FILE. | |
| 002110 | | MOVE SPACES TO PRINT - LINE. | |
| 002120 | | MOVE 'SALARY REPORT FOR EMPLOYEE UNDER 30' | |
| 002130 | | TO PRINT - LINE. | |
| 002140 | | WRITE PRINT - LINE AFTER ADVANCING 2 LINES. | |
| 002150 | | MOVE SPACES TO PRINT - LINE. | |
| 002160 | | WRITE PRINT - LINE AFTER ADVANCING 1 LINE. | |
| 002170 | * | MAIN - LOOP ROUTINE. | |
| 002200 | | READ CARD - FILE | |
| 002210 | | AT END MOVE 'NO' TO MORE - INPUT. | |
| 002212 | | <div style="border: 1px solid black; width: 50px; height: 15px; display: inline-block;"></div> PROCESS - EMPLOYEE - RECORDS | 8 |
| 002220 | | UNTIL MORE - INPUT = 'NO'. | |
| 002221 | * | TERMINATING ROUTINE. | |
| 002222 | | CLOSE CARD - FILE | PRINT - FILE. |

```

002230      STOP RUN.
002231  *      SUBROUTINES
002290  PROCESS - EMPLOYEE - RECORDS.
002291      IF      CARD - AGE < 30
002292          MOVE  TO      PRINT - LINE
002293          MOVE CARD - NAME TO      PRINT - NAME
002294          MOVE  TO      PRINT - AGE
002295          MOVE CARD - SALARY TO      PRINT - SALARY
002296          WRITE PRINT - LINE AFTER ADVANCING 2 LINES.
002297      READ CARD - FILE
002300          AT END MOVE  TO MORE - INPUT.

```

تمرين ٩ :

ادخلت المعلومات الخاصة باستهلاك المشتركين في شركة الكهرباء على بطاقات مشقبة بمعدل بطاقة لكل مشترك واسم الملف CARD - F وفُرِزَت هذه البطاقات تصاعدياً على رقم المشترك علماً بأن شكل سجل الإدخال كما يلي :

| | Field name | Picture |
|-----------------|------------|---------|
| رقم المشترك | C - NO | 9 (5) |
| القراءة الحالية | C - READ | 9 (7) |
| تاريخ القراءة | C - DATE | 9 (8) |
| ملاحظات | C - COMNT | X (30) |
| فراغ | C - FLR | X (30) |

والمطلوب : كتابة برنامج يقرأ الملف CARD - F و ينقل محتواه الى ملف موجود على قرص ممغنت اسمه CONSMFPL وشكله كالتالي :

| <u>Field name</u> | <u>Picture</u> |
|-------------------|----------------|
| CONS - NO | 9 (5) |
| CONS - READ | 9 (9) |
| CONS - DATE | 9 (8) |
| CONS - COMNT | X (3φ) |
| FILLER | X (3φ) |

تمرين ٢ :

في أحد برامج معهد الادارة تقدم للامتحان ١٠٠ متدرب، وبعد ظهور النتائج تبين أن عدد الناجحين ٧٠ متدرباً وعدد الراسبين ٣٠ متدرباً.

ادخلت بيانات المتدربين على شريط ممغنت بعد أن فرزت SORTED بحيث كانت سجلات الناجحين أولاً ثم سجلات الراسبين.

والمطلوب : كتابة برنامج بلغة كوبرول ينقل الطلاب الناجحين على ملف اخراج اسمه DFILE موجود على قرص ممغنت، وينقل الطلاب الراسبين على ملف اخراج آخر اسمه TFILE على قرص ممغنت، مع ملاحظة ما يلي :

- ١ - طول سجل الإدخال هو ٨٠ حرف
- ٢ - استعمل تعليمة PERFORM لحل هذا البرنامج.
- ٣ - اسم ملف الإدخال هو INFILE.

تمرين ٣ :

لديك ملف اسمه STCKCARD موجود على بطاقات مثقبة وشكل السجل فيه

كالتالي :

| | <u>Field name</u> | <u>Picture</u> |
|------------|-------------------|----------------|
| الكمية | QTY | 9 (4) V99 |
| سعر الوحدة | PRICE | 9 (3) V99 |
| غير مستخدم | FILLER | X (96) |

المطلوب كتابة برنامج يحسب السعر الاجمالي للسلعة وذلك بضرب الكمية بسعر الوحدة، ثم طباعة الكمية وسعر الوحدة والسعر الاجمالي على الطباعة ، وفي نهاية العمل اطبع مجموع السعر الاجمالي.



لغة كوبيول المتقدمة

التعليمات المتقدمة في لغة كوبيول
(قسم الاجراءات)

الفصل الرابع

أقسام برنامج كوبيول الأخرى والعبارات
المتقدمة

الفصل الخامس

الجدول في لغة كوبيول

الفصل السادس

الفرز والدمج في لغة كوبيول

الفصل السابع

الملفات ومعالجتها

الفصل الثامن

البرمجة الهيكلية بلغة كوبيول

الفصل التاسع

تدقيق البرنامج واختباره

الفصل العاشر

الفصل الرابع

التعليمات المتقدمة في لغة كوبول (قسم الاجراءات)

٤ - ١ ملحة عامة :

هذا الفصل هو الفصل الأول من القسم الثاني الذي يحتوي على توسعة وتعمق أكثر في تعليمات لغة كوبول التي مرت في القسم الأول وكذلك على بعض التعليمات الجديدة التي تُكَمِّل التعليمات السابقة لتعطي المبرمج قدرة أكبر على كتابة البرنامج واختيار أفضل الحلول .

ففي هذا الفصل سوف يتم ذكر تفاصيل أكثر عن تعليمات المقارنة IF والاختبارات المختلفة المتعلقة بها وكذلك المقارنة المركبة والمتداخلة، ثم ترد تعليمات فحص البيانات INSPECT مع الإشارة الى التعليمات المماثلة EXAMINE، ثم إضافة عبارة التطابق CORRESPONDING على التعليمات الحسابية وتعليمات النقل . كما سوف يتم شرح تعليمات عرض البيانات DISPLAY وقبولها ACCEPT واستخدام INTO مع القراءة و FROM مع الكتابة وعبارتي ROUNDED و ON SIZE ERROR وأخيراً يتم بيان تعليمات GO TO المحسوبة (أو المشروطة).

٤ - ٢ تعليمات IF

وتشتمل هذه التعليمات كما مر معنا المقارنة بين طرفين إضافة إلى شكلها المذكور آنفاً تشمل المقارنات التالية :

٤-٢-١ اختبار نوعية بيانات الحقل CLASS TEST

يحتاج المبرمج أحياناً لتحديد نوعية البيانات الموجودة في الحقل لذلك فقد زودت لغة كوبيول بإمكانية المقارنة على هذا الحقل ومعرفة نوع البيانات وتأخذ تعليمة IF عندئذ الشكل العام التالي :

$$\text{IF identifier} \quad \text{IS} \quad [\text{NOT}] \quad \left\{ \begin{array}{l} \text{NUMERIC} \\ \text{ALPHABETIC} \end{array} \right\}$$

ويجب وصف الحقل المختبر على أنه رقمي بالصورة الرقمية (أي نوع 9 في PIC) ووصف الحقل المراد اختباره على أساس أبجدي ALPHABETIC بالوصف ٥A . كما يمكن إجراء الاختبار على الحقل الموصوف بالنوع X على كلا الحالتين ، والجدول التالي يوضح هذه الصورة بشكل أكثر :

| <u>Data Type</u> | <u>Valid Tests</u> |
|------------------|--|
| Numeric (9) | NUMERIC, NOT NUMERIC |
| Alphabetic (X) | ALPHABETIC, NOT ALPHABETIC |
| | NUMERIC, NOT NUMERIC, ALPHABETIC, NOT ALPHABETIC |

أمثلة : إذا كان لدينا الحقول الثلاثة التالية والموصوفة في ثلاثة أنواع من البيانات :

| | | |
|---|-------------------|-------------|
| 5 | NUMERIC - FLD | PIC 9 (6). |
| 5 | ALPHA - FLD | PIC A (10). |
| 5 | ALPHA - NUM - FLD | PIC X (5). |

(٥) في معظم المترجمات يشترط استخدام عبارة USAGE مع DISPLAY في توصيف الحقل المراد اختباره بعد عبارة .PICTURE

فإنه يمكن كتابة تعليمة الاختبار IF على النحو التالي :

IF NUMERIC - FLD IS NUMERIC
 IF NUMERIC - FLD IS NOT NUMERIC
 IF ALPHA - NUM - FLD IS NOT ALPHABETIC
 IF ALPHA - NUM - FLD IS NUMERIC

ولكن لا يمكن كتابة التعليمات التالية لعدم صلاحيتها :

IF ALPHA - FLD IS NOT NUMERIC
 IF NUMERIC - FLD IS NOT ALPHABETIC

ويجب التنويه أخيراً إلى أن وجود الإشارة الجبرية في الحقل الرقمي لا يسمح باختباره على أنه رقمي .

٤ - ٢ - ٢ اختبار الإشارة الجبرية SIGN TEST

إن هذا النوع من الاختبار بتعليمة IF يحدد إشارة الحقل الرقمي، والشكل العام يأخذ الصورة التالية :

$$\text{IF} \left\{ \begin{array}{l} \text{identifier} \\ \text{arithmetic} \\ \text{expression} \end{array} \right\} \left\{ \begin{array}{l} \text{IS } \underline{\text{[NOT] POSITIVE}} \\ \text{IS } \underline{\text{[NOT] NEGATIVE}} \\ \text{IS } \underline{\text{[NOT] ZERO}} \end{array} \right\}$$

وتكون قيمة الحقل identifier موجبة Positive إذا كان أكبر من الصفر وتكون سالبة إذا كانت أقل من الصفر. ويستخدم هذا الاختبار للتحقق من قيمة البيانات المدخلة أو قيمة البيانات المحسوبة الناتجة عن عملية حسابية.

أمثلة :

- 1) IF NET - PAY IS NOT POSITIVE PERFORM CAL - TAXES.
- 2) IF BALANCE IS NEGATIVE PERFORM OVER - DUE.

٤ - ٢ - ٣ تعريف واختبار اسم الشرط

إن عبارة الشرط CONDITION في تعليمة IF غالباً ما تختبر قيمة الرمز الداخل الى الحاسب (أو إلى البرنامج) مثل :

IF YR - CODE = " I " ...

إذا كان مثل هذا الرمز مسموح به ، إلا أن معنى هذا الرمز في الحقل YR - CODE لا يكون واضحاً وغير معروف في البرنامج . لهذا تضمنت لغة كوبول بديلاً لهذا الشرط يدعى اسم الشرط ليعطي هذه اللغة قوة أكبر في التوثيق والايضاح . وهذا الاسم يظهر في المستوى Level رقم 88 ويعيد توصيف الحقل وتحديد القيمة الممكن أن ترد فيه دون أن يعطي للحقل أي حجم أو وصف آخر . كما يجب استعمال هذا التعريف فقط مع الحقول الأولية elementary أي التي تحتوي على عبارة (PIC) .

مثال :

05 YR - CODE PIC X.

```
88 GEORGIAN      VALUE " 1 ".
88 HAGRI         VALUE " 2 ".
88 OTHER         VALUE " 3 ".
88 VALID - YR - CODE VALUE " 1", " 2", " 3".
```

فإذا وجدت هذه التوصيفات في قسم وصف البيانات DATA DIVISION فإن المقارنة تصبح كما يلي :

IF GEORGIAN PERFORM M - S.

وهي تعادل :

IF YR - CODE = "1" PERFORM M - S.

وهناك ثلاثة فوائد من اسم الشرط وهي :

- زيادة في الايضاح وتوثيق البرنامج .
- تسهيل تعديل وتطوير البرنامج وامكانية إضافة رموز أخرى على الحقل دون إجراء تعديل كبير على البرنامج .
- السماح بتجميع عدة رموز بتعريف واحد في المستوى 88 .

وبشكل أكثر تفصيلاً يمكن وضع الصيغة العامة لتعريف اسم الشرط على النحو التالي :

$$88 \text{ data - name } \left\{ \begin{array}{c} \underline{\text{VALUE IS}} \\ \underline{\text{VALUES ARE}} \end{array} \right\} \text{ literal - 1 [THRU literal - 2]} \\ \text{[literal - 3 [THRU literal - 4]]....}$$

مثال :

| 88 | NATION - CODE | PIC 9. |
|----|---------------|------------------------|
| 88 | SAUDI | VALUE 1. |
| 88 | ENGLISH | VALUE 2. |
| 88 | TURKISH | VALUE 3. |
| 88 | PAKISTAN | VALUE 4. |
| 88 | AFRICAN | VALUES ARE 5 THRU 10. |
| 88 | AMERICAN | VALUES ARE 20 THRU 26. |
| 88 | AUSTRALIAN | VALUES ARE 30 , 33. |

IF ENGLISH GO TO E - RTN. وتعليمة :

IF NATION - CODE = 2 GO TO E - RTN. تعادل :

٤ - ٢ - ٤ المقارنة المركبة COMPOUND IF

يمكن جمع أكثر من شرط (مقارنة) بتعليمة IF حيث تشكل فيما بينها مقارنة مركبة و يستخدم لهذا الغرض المعاملات المنطقية AND و OR و NOT. وعندئذ يمكن أن تأخذ تعليمة IF المركبة الشكل العام التالي :

- 1)
$$\text{IF Condition - 1 } \left\{ \left[\left[\begin{array}{c} \text{AND} \\ \text{OR} \end{array} \right] \text{ Condition - 2 } \right] \dots \right\} \left\{ \begin{array}{l} \text{Statement - 1} \\ \text{NEXT SENTENCE} \end{array} \right\} \left[\text{ELSE} \left\{ \begin{array}{l} \text{Statement - 2} \\ \text{NEXT SENTENCE} \end{array} \right\} \right]$$
- 2)
$$\text{IF NOT Condition } \left\{ \begin{array}{l} \text{Statement - 1} \\ \text{NEXT SENTENCE} \end{array} \right\} \left[\text{ELSE} \left\{ \begin{array}{l} \text{Statement - 2} \\ \text{NEXT SENTENCE} \end{array} \right\} \right]$$

حيث يقصد بكلمة : Condition الشرط ويمكن أن يكون على النحو التالي ٥٥ :

- مقارنة بسيطة Simple Condition
- شرط بسيط منفي Negated Simple Condition
- شرط مركب Compound Condition
- شرط مركب منفي Negated Compound Condition
- مجموعة من الشروط البسيطة أو المركبة

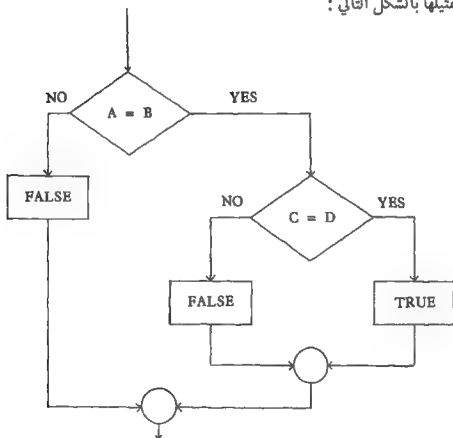
(٥) Logical Operators

(٥٥) T1990 COBOL

فالعامل المنطقي AND يعني أنه إذا تحقق الشرط الأول والثاني البسيطين أو المركبين في تعليمة IF فإن النتيجة للمقارنة الكلية تكون (True) محققة .
فالمقارنة المركبة التالية :

IF A = B AND C = D

يمكن تمثيلها بالشكل التالي :



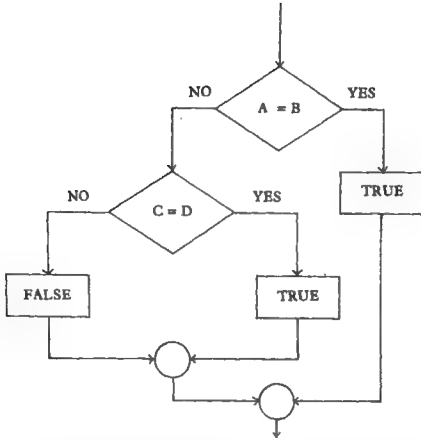
وإذا لم يتحقق أحد الشرطين (A = B) أو (C = D) فإن نتيجة IF تكون (False) غير محققة .

أما العامل المنطقي OR فيعني أن تحقق أحد الشرطين المذكورين في تعليمة IF يكفي لتكون نتيجة المقارنة True .

فالمقارنة المركبة :

IF $A = B$ OR $C = D$

يمكن تمثيلها بالشكال التالي :



ويعوجب استخدام العامل OR فان تحقق أحد الشرطين $(A = B)$ أو $(C = D)$ يؤدي الى الحصول على النتيجة الصحيحة True في الرسم وعدم تحقق كليهما معاً ينفي تحقق النتيجة المرجوة من المقارنة .

أما العامل المنطقي NOT فهو إذا أضيف الى تعليمة IF فيكون عدم تحقق الشرط هو النتيجة المطلوبة من المقارنة True وتحقق الشرط يؤدي الى نقل التسلسل إلى جهة . False

فمثلاً تعليمة IF التالية توضح ذلك :

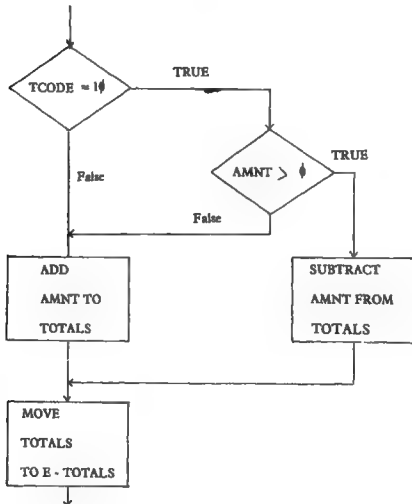
IF NOT (TCODE IS = 10 AND AMNT IS > 0)

ADD AMNT TO TOTALS

ELSE SUBTRACT AMNT FROM TOTALS -

MOVE TOTALS TO E - TOTALS.

والخطط الانسيابي التالي يبين تسلسل تنفيذ هذه التعليمات :



و يلاحظ عند استخدام المعاملات المنطقية امكانية استخدام الأقواس من أجل تحديد المقارنات والشروط في تعليمة IF لأن أولوية تنفيذها تلعب دوراً كبيراً عند عدم وجود الأقواس حيث أن ترتيب تنفيذ العمليات المختلفة في تعليمة IF الواحدة تكون وفق التسلسل التالي :

- ١ - الشروط البسيطة (أو المعاملات العلائقية) Relational Operators.
- ٢ - العامل المنطقي NOT
- ٣ - العامل المنطقي AND (ويبدأ من جهة اليسار إلى اليمين في حال وجود أكثر من معامل).
- ٤ - العامل المنطقي OR (ويبدأ من جهة اليسار إلى اليمين في حال وجود أكثر من معامل).

أمثلة :

التعليمات التالية تبين استخدام AND مع OR :

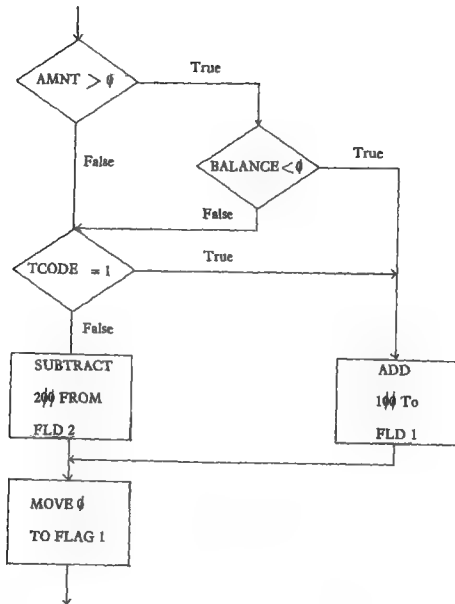
```
IF   TCODE = 1   OR AMNT > 0 AND BALANCE < 0
      ADD  1000  TO FLD 1
```

ELSE

SUBTRACT 2000 FROM FLD 2.

MOVE 0 TO FLAG 1.

وفيما يلي المخطط الانسيابي لهذه التعليمات :



والتعليمات التالية تبين استخدام AND مع OR بوجود أقواس لحصر الشرط
المستخدم مع OR :

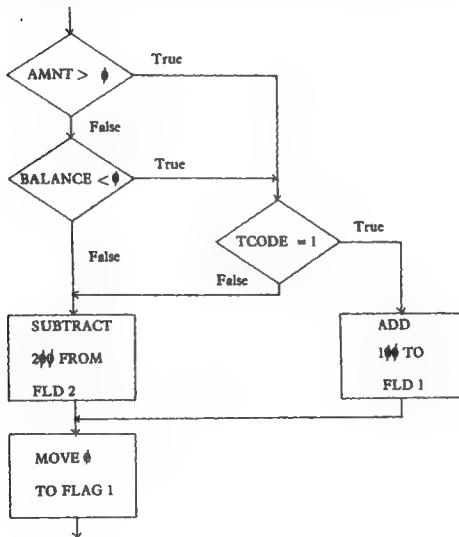
IF TCODE = 1 AND (AMNT > 0 OR BALANCE < 0)
ADD 100 TO FLD 1

ELSE

SUBTRACT 200 FROM FLD 2.

MOVE 0 TO FLAG 1.

ويعين أيضاً المخطط الانسيابي التالي شكل التسلسل المنطقي لتنفيذ هذه التعليمات مع ملاحظة الفرق بين هذا المثال والمثال السابق بسبب وجود الأقواس .



٤ - ٢ - ٥ تعليمة IF المتداخلة NESTED IF

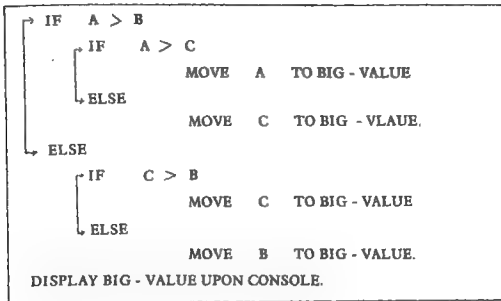
إن الشكل العام لتعليمة IF هو ما يلي :



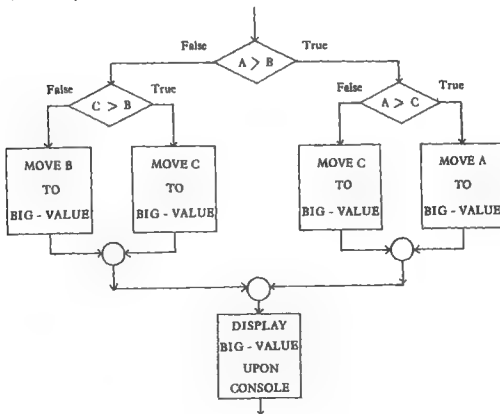
حيث أن الشرط Condition يمكن أن يكون أي نوع من المقارنة (الاختبارات Tests) مثل مقارنة النوع Class أو اسم الشرط Condition - name أو مقارنة بسيطة أو مقارنة الإشارة الجبرية أو مقارنة مركبة . وعبارة NEXT SENTENCE يقصد بها متابعة تنفيذ البرنامج بعد النقطة الموجودة في آخر تعليمة IF .

أما تعليمة IF المتداخلة (المتشابهة) فهي عبارة عن كون أي من العبارات Statement - 1 أو Statement - 2 بحد ذاتها تعليمة IF أخرى . أو بمعنى آخر هناك أكثر من تعليمة IF في جملة واحدة .

ويجب على المبرمج ملاحظة أن كل تعليمة IF يتبعها عبارة ELSE الخاصة بها وتكون كل كلمة ELSE في الجملة تابعة لأقرب IF لها حسب قاعدة الترجمة Compilation بلغة كوبول . وعندما لا يوجد لكل IF عبارة ELSE فيجب على المبرمج تحديد النهاية المنطقية لـ IF لمعرفة تبعيتها لأي عبارة ELSE . ولأجل سهولة الكتابة والمراجعة تكتب تعليمة IF المتداخلة على شكل مدرج من جهة اليسار لتحديد تسلسلها المنطقي وتحديد عبارة ELSE التابعة لها . والشكل التالي يعطي مثالا على ذلك :



ويمكن رسم هذه التعليمات التي تقوم باختيار أكبر قيمة من بين ثلاث قيم على التوالي:

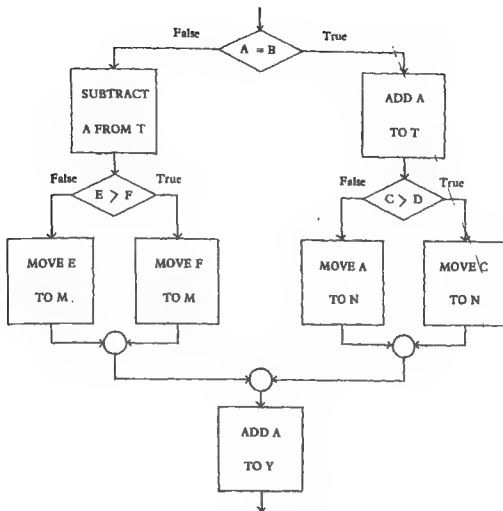


أما التعليمات التالية فهي تمثل مثال آخر على استخدام IF المتداخلة :

```
IF A = B ADD A TO T
  IF C > D MOVE C TO N
  ELSE
    MOVE A TO N
ELSE
  SUBTRACT A FROM T
  IF E > F MOVE F TO M
  ELSE
    MOVE E TO M.
ADD A TO Y.
```

و يلاحظ من المثالين أن جملة IF المتداخلة تنتهي بنقطة واحدة Period في نهاية آخر مقارنة ويجب على المبرمج الانتباه الشديد لوجود أية نقطة أخرى ضمن المقارنات المتداخلة في الجملة الواحدة لأنها سوف تسبب خللاً في منطق تسلسل المقارنات. ويستعان عادة بالمخطط الانسيابي لتوضيح سير المقارنات و يشار الى الجهة التي تحقق الشرط الموضوع في المقارنة بكلمة True أو Yes وإلى الجهة الأخرى (عدم تحقق الشرط) بكلمة False أو No.

والرسم التالي يبين سير المخطط الانسيابي للتعليمات السابقة الذكر.



٤-٣ استخدام عبارة CORRESPONDING

في الحالات العادية يوصف المبرمج أسماء المتغيرات والحقول والسجلات في برنامجه بشكل لا يسمح بتكرار الاسم الواحد لأكثر من مرة أي أن أسماء المتغيرات تكون فريدة UNIQUE، ولكن مع تطور لغة كوبيول وظهور النسخ الجديدة منها أدى الى السماح بتكرار أسماء الحقول فقط ضمن السجلات التي تحمل أسماء مختلفة وذلك من أجل زيادة فعالية هذه اللغة في نقل الحقول من سجل لآخر ولتوفير الجهد في كتابة

تعليمة MOVE بشكل خاص. فمثلاً لو كان لدينا السجلان التاليان في قسم البيانات :

01 CARD - REC.

| | | | |
|----|----------------|-----|---------|
| 05 | STD - NAME | PIC | (25). |
| 05 | STD - ADDRESS. | | |
| 10 | STREET | PIC | X (15). |
| 10 | CITY - NAME | PIC | A (15). |
| 05 | STD - SCORES | PIC | 999. |
| 05 | STD - GRADE | PIC | X. |
| 05 | FILLER | PIC | X (21). |

01 .

.
.
.
.

01 PRINT - LN.

| | | | |
|----|----------------|-----|---------|
| 03 | STD - NAME | PIC | A (25). |
| 03 | FILLER | PIC | X (4). |
| 03 | STD - ADDRESS. | | |
| | 07 STREET | PIC | X (15). |
| | 07 CITY - NAME | PIC | A (15). |
| 03 | FILLER | PIC | X (3). |
| 03 | STD - GRADE | PIC | X. |
| 03 | FILLER | PIC | X (3). |
| 03 | STD - SCORES | PIC | 999. |

| | | | |
|----|------------------|-----|---------|
| Ø3 | FILLER | PIC | X (5). |
| Ø3 | SPECIAL - NUMBER | PIC | 9999. |
| Ø3 | FILLER | PIC | X (54). |

فإن لغة كوبرول تسمح بوجود نفس الاسم في السجلين للحقل الواحد، ولكن يجب الإشارة الى تبعية هذا الحقل عند استخدامه في قسم الاجراءات فمثلاً لو أردنا جمع الحقل STD - SCORES فيجب الإشارة الى تبعية في تعليمة الجمع ADD :

ADD STD - SCORES IN CARD - REC TO TOTAL - SCORES.

وتستخدم للدلالة على التبعية Qualification كلمة IN أو كلمة OF مثل :

MOVE STD - GRADE OF CARD - REC TO STD - GRADE OF

PRINT - LN.

ويمكن أن تكون التبعية على أكثر من مستوى مثل :

MOVE STREET OF STD - ADDRESS OF CARD - REC TO STREET IN

STD - ADDRESS IN PRINT - LN.

كما أنه يمكن الإشارة الى تبعية الأصلية في السجل فقط فيكون المثال السابق :

MOVE STREET IN CARD - REC TO STREET IN PRINT - LN

وهذا يعادل نفس النتيجة المذكورة أعلاه.

واختيار نفس الأسماء في أكثر من سجل يوفر على المبرمج الوقت اللازم لايجاد أسماء فريدة لكل حقل من الحقول، كما يساعد في نقل أو تداول هذه الحقول بشكل مختصر أكثر عند استخدام كلمة CORRESPONDING أو CORR في التعليمة. وتكون

تعليلة النقل MOVE عندئذ على النحوالتالي :

MOVE { CORRESPONDING } identifier - 1 TO identifier - 2.
CORR

ويلاحظ أن كلمة CORR هي اختصار لكلمة CORRESPONDING . وباستخدام هذا الشكل لتعليلة النقل MOVE نستطيع نقل (أو نسخ) الحقول في السجل CARD - REC الى الحقل في السجل PRINT - LN بتعليلة واحدة . أي :

MOVE CORR CARD - REC TO PRINT - LN.

حيث تقوم بنقل جميع محتويات الحقول التي تحمل نفس الاسم في السجل CARD - REC الى الحقول نفسها في السجل PRINT - LN . وهذه التعليلة تعادل التعليلات التالية :

MOVE STD - NAME IN CARD - REC TO STD - NAME IN PRINT - LN.

MOVE STREET IN CARD - REC TO STREET IN PRINT - LN.

MOVE CITY - NAME IN CARD REC TO CITY - NAME IN
 PRINT - LN.

MOVE STD - SCORES IN CARD - REC TO STD - SCORES IN
 PRINT - LN.

MOVE STD - GRADE IN CARD - REC TO STD - GRADE IN
 PRINT - LN.

وتجب الاشارة الى أن الحقول التي ليست مشتركة في الاسم بين السجلين تستثنى من عملية النقل . مثل الحقل SPECIAL - NUMBER .

و يستخلص من عبارة التطابق الاختيارية النقاط التالية عند استخدامها :

١ - يتم النقل بين السجلين بواسطة MOVE للحقول الأولية elementary ويجب أن يكون هناك في السجلين حقل واحد متطابق في كل منهما .

٢ - يتم النقل للحقول المتطابقة في الاسم بين السجلين بغض النظر عن ترتيب الحقل في السجل و يعتبر اسم الحقل معيار التطابق .

٣ - الحقول المتطابقة في الاسم يجب أن لا تحتوي العبارات التالية في توصيفها :

REDEFINES , RENAMES , OCCURS , USAGE IS INDEX.

لأنها عندئذ لا يتم نقلها بعبارة التطابق مع MOVE .

وإضافة الى إمكانية استعمال كلمة CORR مع تعليمة MOVE فإنه يمكن استعمالها بنفس الأسلوب مع تعليمة الجمع ADD وكذلك الطرح SUBTRACT .

و يكون الشكل العام لهذين التعليمتين عندئذ كما يلي :

ADD { CORRESPONDING } identifier - 1 TO identifier - 2
CORR
[ROUNDED] [ON SIZE ERROR imp - statement]

SUBTRACT { CORRESPONDING } identifier - 1 FROM identifier - 2
CORR
[ROUNDED] [ON SIZE ERROR imp - statement]

(٥) GRAUER AND CRAWFORD, PP 118.

(٥٥) ينصح بالرجوع لترجم كربول التوفر للحاسب المستخدم لمعرفة شمول عبارة التطابق لهذه التعليمات .

٤ - ٤ تعليمة القراءة READ INTO

يمكن كتابة تعليمة القراءة READ التي سبق ذكرها في الشكل التالي العام :

READ file - name RECORD INTO identifier AT END imperative statement.

واضافة كلمة INTO على تعليمة القراءة تمكن المبرمج من قراءة السجل من الملف الموصوف في هذه التعليمة ونقل محتويات السجل الى المكان identifier الموصوف في قسم التوصيف الداخلي للبرنامج أي WORKING - STORAGE SECTION . والنال يوضح استخدام هذه التعليمة :

FD CARD - FILE ...

01 CARD - REC PIC X (80).

WORKING - STORAGE SECTION.

01 WORK - REC - AREA PIC X (80).

PROCEDURE DIVISION.

READ CARD - FILE INTO WORK - REC - AREA AT END

PERFORM END - OF - READ.

وبعد تنفيذ تعليمة القراءة تكون محتويات السجل في الحقل WORK - REC - AREA أي أن هذه التعليمة تشمل تعليمة النقل MOVE .

Ø1 HEADER - LINE - 1.

Ø5 FILLER PIC X (30) VALUE SPACE.

PROCEDURE DIVISION.

WRITE P-REC FROM HEADER - LINE - 1 AFTER ADVANCING
T - OF - PG LINES.

وهذه التعليمة WRITE FROM تعادل التعليمتين التاليتين :

MOVE HEADER - LINE - 1 TO P - REC.

WRITE P - REC AFTER T - OF - PG LINES.

٤ - ٦ التقريب وكشف الخطأ في التعليمات الحسابية

ROUNDED and ON SIZE ERROR CLAUSES

إن التقريب في الأعداد الحسابية يعتبر ضرورياً خاصة في الأعمال التجارية لتتوافق مع الاجزاء النقدية الصغيرة ، لذلك احتوت لغة الكوبول على امكانية التقريب باضافة كلمة ROUNDED على التعليمات الحسابية المعروفة في تعليمة الجمع أو الطرح أو الضرب أو التقسيم أو تعليمة COMPUTE . و يكون موضع هذه الكلمة في التعليمة الحسابية الجمع أو الطرح أو الضرب أو القسمة كما يلي :

$$\left\{ \begin{array}{l} \text{arithmetic} \\ \text{Statement} \end{array} \right\} \left[\underline{\text{ROUNDED}} \right]$$

وعند وجود كلمة الباقي في التقسيم REMAINDER فانها تأتي بعد كلمة

ROUNDED. أي : DIVIDE [ROUNDED]

[REMAINDER] identifier.

أما في تعليمة COMPUTE فتأتي على النحو التالي :

COMPUTE identifier - 1 [ROUNDED] = arithmetic expression

وبكلمة ROUNDED تكون النتيجة الحسابية التي تحتوي على جزء عشري أكثر دقة لأنها تؤدي الى التقريب في النتيجة الى أقرب موضع في الجزء العشري حسب قواعد التقريب في التعليمات الحسابية العادية. أي أن الرقم العشري الأكثر من ٤ يقرب الى القيمة ١ والرقم العشري المساوي أو أقل من ٤ يقرب الى القيمة صفر.

وليكن لدينا الحقلان 1 - FLD و 2 - FLD موصوفين على النحو :

05 FLD - 1 PIC 99V999 VALUE 2.857

05 FLD - 2 PIC 99V999 VALUE 15.142

والحقل 3 - FLD موصوف على النحو :

05 FLD - 3 PIC 99V99

ثم نفذت تعليمة الجمع ADD المكتوبة وفق الشكل :

ADD FLD - 1, FLD - 2 GIVING FLD - 3 ROUNDED.

فتكون نتيجة الجمع 18.00 بدلا من القيمة 17.99

أما من أجل كشف الخطأ في العمليات الحسابية فانه يمكن استخدام الجملة ON SIZE ERROR مع التعليمات الحسابية المعروفة في لغة كوبيول ويمكن هذه الجملة المبرمج التأكد من أن الحقل الذي سوف يخزن فيه (يحفظ) النتيجة ذو حجم كافي لها أو

أنها تزيد عن عدد البايت المحجوزة (المعرفة) في قسم البيانات. وهذه الجملة في التعليمات الحسابية تقوم بمقام التعليمة IF في حال حصول عدم تطابق في حجم النتيجة المحسوبة وحجم الحقل فان تسلسل البرنامج ينتقل الى التعليمة الأمرية Imperative Statement الموصوفة بعد هذه الجملة من أجل اعطاء المبرمج فرصة التأكد من صحة حساباته. ويكون موضع هذه الجملة في التعليمات الحسابية الجمع أو الطرح أو الضرب أو التقسيم على الشكل التالي :

$$\left\{ \begin{array}{l} \text{arithmetic} \\ \text{statement} \end{array} \right\} \quad [\text{ON SIZE ERROR}] \quad \left\{ \begin{array}{l} \text{imperative} \\ \text{statement} \end{array} \right\}$$

وفي حال اجتماع جملة التقريب مع جملة كشف الخطأ في حجم الحقل تأخذ التعليمات الحسابية الشكل التالي :

$$\left\{ \begin{array}{l} \text{arithmetic} \\ \text{statement} \end{array} \right\} \quad \underline{\text{ROUNDED}} [\text{ON SIZE ERROR}] \quad \left\{ \begin{array}{l} \text{imperative} \\ \text{statement} \end{array} \right\}$$

أما تعليمة التقسيم فيمكن أن تكون على النحو التالي :

$$\underline{\text{DIVIDE}} \dots [\text{ROUNDED}] [\underline{\text{REMAINDER}} \text{ identifier}] [\text{ON SIZE ERROR}] \quad \left\{ \begin{array}{l} \text{imperative} \\ \text{statement} \end{array} \right\}$$

بينما تعليمة احسب COMPUTE تصبح حسب الشكل العام التالي :

$$\text{COMPUTE identifier} - 1 [\text{ROUNDED}] = \text{arithmetic expression} \dots$$

$$[\text{ON SIZE ERROR}] \quad \left\{ \begin{array}{l} \text{imperative} \\ \text{statement} \end{array} \right\}$$

فليكن لدينا الحقول التالية :

| | | |
|----|---------------|------------------|
| 05 | HOURLY - RATE | PIC 99 VALUE 25. |
| 05 | HOURS | PIC 99 VALUE 40. |
| 05 | GROSS - PAY | PIC 999. |

وتعليمة احسب التالية :

COMPUTE GROSS - PAY = HOURLY - RATE * HOURS

ON SIZE ERROR

PERFORM ERROR - RTN.

فان عملية ضرب الحقلين تعطي النتيجة 1000 ولكن الحقل GROSS - PAY معرف على أنه ثلاثة أرقام فقط (PIC 999) ويخزن فيه القيمة 000 وهذه النتيجة غير صحيحة حسابياً لذلك فان جملة ON SIZE ERROR توجه سير تسلسل البرنامج الى الروتين المذكور بعدها والمعروف من قبل المبرمج ليأخذ الاحتياطات اللازمة في مثل هذه الحالة .

٤ - ٧ تعليمة نقل التسلسل المحسوبة GO TO DEPENDING ON

عرفنا في السابق تعليمة نقل التسلسل البسيطة GO TO التي تقوم بنقل تسلسل تنفيذ البرنامج الى الفقرة المذكورة بعدها . ولكن اذا كان لدينا عمليات نقل تسلسل نعتد على قيمة معينة مثل :

IF CODE - S = 1 GO TO RTN - A.

IF CODE - S = 2 GO TO RTN - B.

IF CODE - S = 3 GO TO RTN - C.

GO TO ERROR - RTN.

حيث أن عملية نقل التسلسل الى الفقرات RTN - A و RTN - B و RTN - C تعتمد على قيمة الحقل CODE - S ولتسهيل عملية البرمجة في كوبيول فقد احتوت اللغة على تعليمة GO TO مع عبارة (جملة) DEPENDING ON ويمكن كتابة المثال السابق كما يلي :

GO TO RTN - A , RTN - B , RTN - C DEPENDING ON CODE - S.

GO TO ERROR - RTN.

حيث أن القيمة «1» في الحقل CODE - S تشير الى اسم الفقرة الأولى RTN - A و «2» تشير الى اسم الفقرة RTN - B وهكذا ... وإذا كانت قيمة الحقل CODE - S مساوية للصفر أو سالبة أو أكبر من عدد الفقرة المقررة بعد GO TO ينتقل تسلسل التنفيذ الى الجملة التالية أي :

GO TO ERROR - RTN.

وتأخذ تعليمة نقل التسلسل المحسوبة الشكل العام التالي :

GO TO paragraph - 1, paragraph - 2 paragraph - 3 paragraph - name - n

DEPENDING ON data - name.

حيث أن paragraph - i تشير الى اسم الفقرة المراد نقل التسلسل اليها و data - name اسم الحقل الرقمي الذي يجب أن يحتوي على قيمة صحيحة integer تدل على اسماء الفقرات المذكورة بعد GO TO .
والحد الأعلى لعدد الفقرات يختلف باختلاف نوع الحاسب المستخدم إلا أنه يتجاوز ١٠٠٠ فقرة .

(*) Stern Stern, STRUCTURED COBOL PROGRAMMING P. 320.

بعض الحاسبات يصل الى ٢٠٣١ فقرة .

٤ - ٨ تعليمة عرض البيانات DISPLAY

إن تعليمة عرض البيانات DISPLAY تتيح للمبرمج إظهار بيانات بدون تحديد توصيف لها في قسم البيانات ولا تحتاج الى توصيف ملف اخراج OUTPUT في الجزء الخاص بتعريف الملفات والشكل العام لهذه التعليمة حسب قواعد ANSI هو :

$$\text{DISPLAY} \left\{ \begin{array}{l} \text{identifier - 1} \\ \text{literal - 1} \end{array} \right\} \left[\left[\begin{array}{l} \text{identifier - 2} \\ \text{literal - 2} \end{array} \right] \right] \text{-----}$$

$$\left[\text{UPON mnemonic - name} \right]$$

أمثلة :

- 1) DISPLAY STUDENT - NAME.
- 2) DISPLAY "NAME IS =" STUDENT - NAME.
- 3) DISPLAY STUDENT - NAME, STUDENT - NO.
- 4) DISPLAY FIELD - A UPON CONSOLE

في المثال (١) يتم إظهار محتوى الحقل المسمى STUDENT - NAME على الشاشة لعدم ذكر وتحديد الوحدة المراد استخدامها في إظهار البيانات . أما المثال (٢) يقوم الحاسب بإظهار عبارة "NAME IS =" قبل إظهار محتوى الحقل المسمى STUDENT - NAME. والمثال (٣) يتيح للحاسب إظهار محتوى حقليْن ، وكل هذه البيانات تظهر على الشاشة أيضاً لعدم ذكر رمز الوحدة المطلوبة .

وفي المثال رقم (٤) وباستخدام عبارة UPON الاختيارية يقوم الحاسب بإظهار وعرض البيانات على الوحدة المحددة بعد كلمة UPON ، وهي وحدة CONSOLE .-وحدة التوجيه والتحكم في الحاسب).

ويمكن توصيف وحدات إظهار أخرى مثل I / O TYPEWRITER أو آلة طباعة
فرعية ويجب الرجوع في تحديد رمز هذه الوحدة إلى مراجع الشركة الصانعة للحاسب،
لأنها تختلف باختلاف نوع الحاسب المستخدم.

إضافة إلى الشكل العام المعتمد من قبل هيئة ANSI ولضغوط التعامل مع
الشاشة الضوئية الحديثة بواسطة برنامج كويول يوجد شكل آخر لتعليمية عرض البيانات
وهذا الشكل يعتمد على غالبية الحاسبات المسماة MINI - COMPUTER والتي
ترتبط بشاشات مرئية مباشرة من أجل التشغيل المباشر ON - LINE.

$$\begin{aligned}
 & \text{DISPLAY} \left\{ \begin{array}{l} \text{identifier - 1} \\ \text{literal - 1} \end{array} \right\} \left[\text{UNIT} \left\{ \begin{array}{l} \text{identifier - 2} \\ \text{literal - 2} \end{array} \right\} \right] \\
 & \left[\text{LINE} \left\{ \begin{array}{l} \text{identifier - 3} \\ \text{literal - 3} \end{array} \right\} \right] \left[\text{POSITION} \left\{ \begin{array}{l} \text{identifier - 4} \\ \text{literal - 4} \end{array} \right\} \right] \\
 & \left[\text{SIZE} \left\{ \begin{array}{l} \text{identifier - 5} \\ \text{literal - 5} \end{array} \right\} \right] \left[\text{BEEP} \right] \left[\text{ERASE} \right] \left[\begin{array}{l} \text{LOW} \\ \text{HIGH} \end{array} \right] \\
 & \left[\begin{array}{l} \text{identifier - 6} \\ \text{literal - 6} \end{array} \right] \left[\text{UNIT} \left\{ \begin{array}{l} \text{identifier - 7} \\ \text{literal - 7} \end{array} \right\} \right] \left[\text{LINE} \left\{ \begin{array}{l} \text{identifier - 8} \\ \text{literal - 8} \end{array} \right\} \right] \\
 & \left[\text{POSITION} \left\{ \begin{array}{l} \text{identifier - 9} \\ \text{literal - 9} \end{array} \right\} \right] \left[\text{SIZE} \left\{ \begin{array}{l} \text{identifier - 10} \\ \text{literal - 10} \end{array} \right\} \right]
 \end{aligned}$$

$$[, \underline{\text{BEEP}}] [, \underline{\text{ERASE}}] \left[\left[\begin{array}{c} , \underline{\text{LOW}} \\ , \underline{\text{HIGH}} \end{array} \right] \right] \dots$$

إن الثوابت (... literal - 6 literal - 1) يمكن أن تكون أي ثابت شكلي constant figurative . (أي SPACE, ZERO ...)

إن تعليمة العرض هذه تمكن المبرمج من عرض أي متغير أو ثابت (1 - identifier أو literal - 6 أو identifier - 6 literal ، ...) على الشاشة الضوئية .

عندما تحتوي تعليمة DISPLAY على أكثر من متغير ثابت لأجل عرضها أو إظهارها فإن ذلك يتم على التالي حسب ترتيبهم بعد كلمة DISPLAY .

عبارة UNIT إذا ذكرت فإنها يجب أن تكتب قبل بقية العبارات الأخرى الاختيارية، والقيمة التي بعد كلمة UNIT تُوصف (أو تحدد) رقم الوحدة المستخدمة في إظهار البيانات (أي الشاشة الضوئية التي تعرف أحيانا برقم خاص بها) . وإذا لم تذكر عبارة UNIT فإن عملية الاظهار في التعليمة تتم على نفس الشاشة التي ينفذ عليها البرنامج .

عبارة LINE تحدد رقم السطر المراد استخدامه في عرض البيانات على الشاشة (تحتوي الشاشة عادة على ٢٤ سطراً) . وإذا كانت قيمة رقم السطر أكبر من عدد الأسطر على الشاشة فإن البيانات تظهر على آخر سطر في الشاشة . أما إذا كانت قيمة رقم السطر صفر ZERO أو غير معرفة نهائياً فإن البيانات تظهر في المكان التالي مباشرة لموضع ال CURSOR حين تنفيذ التعليمة . ورقم السطر يمكن زيادته للانتقال الى سطر لاحق وهكذا ... فإذا وصل التسلسل الى آخر سطر في الشاشة ينتقل بعد ذلك الى أول سطر فيها بعد مسحها ERASE آلياً .

وعبارة POSITION تدل على مكان وضع الرمز في السطر الواحد على الشاشة (حيث أن عدد الرموز في السطر يصل الى ٨٠ رمزاً عادةً ، وفي بعض الشاشات قد يكون أقل من ذلك) . والقيمة التي تلي كلمة POSITION يجب أن تكون قيمة عددية صحيحة كما هي الحال في رقم السطر وضمن عدد الأماكن المتاحة على الشاشة . ويبدأ السطر من جهة اليسار بالموضع رقم (١) وينتهي الى جهة اليمين بالموضع رقم (٨٠) وإذا لم تذكر القيمة فإن الحاسب يفترض وجود القيمة (١) . وإذا تم ذكر قيمة العدد (صفر) يتم عرض البيانات في الموضع التالي لوجود CURSOR ، وينتهي العرض عند آخر رمز مراد عرضه .

عبارة SIZE تدل على حجم البيانات المعروضة في الحقل أو الثابت الذي يلي كلمة DISPLAY . والقيمة التي تلي كلمة SIZE تحدد هذا الحجم .

عبارة BEEP تعني عند وصفها في تعليمة DISPLAY إظهار صوت الإشارة الخاصة بالشاشة عند عرض أي رمز من البيانات المطلوب عرضها .

أما وجود عبارة ERASE في تعليمة العرض فيدل على طلب المبرمج مسح أو تنظيف الشاشة كلها قبل عرض البيانات الواردة في تعليمة DISPLAY . وفي حال عدم ذكر هذه العبارة يتم العرض على الشاشة دون عملية تنظيف لها سابقة للعرض .

أما عبارة LOW أو HIGH فهما كلمتان وجود أحدهما يدل على نوع الحروف المراد إظهارها فكلمة LOW تعني LOW intensity وهي تشكل الحروف العادية على الشاشة أما كلمة HIGH فتعني HIGH intensity ويقصد بها الرموز والحروف ذات الاضاءة الأشد عند الإظهار على الشاشة ، وتستخدم هذه العبارة لإظهار الحروف في شكلين من شدة الاضاءة لأجللفت نظر العامل أو مستخدم الشاشة ، وتركيز اهتمامه في نوع البيانات المعروضة .

(٥) يمكن في بعض شاشات IBM أن تصل سعة السطر الواحد الى ١٣٢ رمزاً .

وتقبل التعليمة **DISPLAY** إظهار حقل أو ثابت واحد أو أكثر وتكرر أسماء الحقول أو الثوابت المراد اظهارها مع العبارات الخاصة بمكان وشكل هذه البيانات المطلوب اظهارها على الشاشة ، وهذا ما يُقصد بوجود (...) في نهاية الشكل العام للتعليمة .

٤ - ٩ تعليمة قبول أو استلام البيانات **ACCEPT**

إن تعليمة قبول البيانات من خارج البرنامج الى الذاكرة تقوم مقام تعليمة **READ** للبيانات ولكن ليس من ملف خاص بل من وحدة إدخال بطيئة عادة (سابقا كانت وحدة شبيهة بالآلة الكاتبة تسمى **1/O TYPEWRITER**) وكان لا ينصح في استخدامها من أجل إدخال حجم البيانات الكبير وشكلها العام كما يلي :

ACCEPT identifier [FROM mnemonic - name]

وكما في تعليمة **DISPLAY** فإن تسمية رمز وحدة الإدخال اختياري وإذا لم يذكر فإن عملية قبول البيانات تتم عن طريق قارئ البطاقات **CARD READER** المرتبط بالحاسب وإذا ذكرت عبارة **FROM** فإنها تدل على وحدة الإدخال المرغوبة .

ويوجد هناك شكل آخر لتعليمة استلام البيانات في لغة الكوبول **COBOL 74** تستخدم للحصول على التاريخ الجاري ليوم التشغيل أو يوم التشغيل أو الوقت عند تشغيل البرنامج . وشكل هذه التعليمة هو :

ACCEPT identifier FROM { DATE
DAY
TIME }

وعبارة DATE تعطي التاريخ على أساس ٦ أرقام والرقمان في أقصى اليسار يدلان على السنة الميلادية ثم رقم الشهر ثم رقم اليوم أي أن تاريخ March, 16 1979 يكون على الشكل 790316.

أما إذا اختيرت كلمة DAY فيتوفر لدى المبرمج عدد الأيام منذ اليوم الأول في العام لغاية يوم تشغيل البرنامج (السنة تحتوي على ٣٦٦ يوم). فمثلا يوم March 16, 1979 يظهر على النحو : 79075 بينما يوم March 16, 1980 يظهر على النحو : 80076 على اعتبار سنة ١٩٨٠ سنة كبيسة.

أما عبارة TIME فهي تعطي الوقت بالساعات والدقائق والثواني فإذا كانت الساعة ١٠ و ١٥ دقيقة صباحاً يظهر في الحقل المستلم للوقت : 10150000 أما الوقت الساعة ١٠ و ١٥ دقيقة مساءً فيكون : 22150000 والرقمان الى أقصى اليمين يمثلان الأجزاء المئوية للثانية.

أمثلة :

- 1) ACCEPT NAME - 1 FROM CONSOLE.
- 2) ACCEPT STATE - NO.
- 3) ACCEPT DATE - FLD FROM DATE.
- 4) ACCEPT DAY - NUM FROM DAY.
- 5) ACCEPT TIME - FLD FROM TIME.

وبعد تطور صناعة الحاسبات وإدخال الشاشات الضوئية في حقل إدخال البيانات والتعامل المباشر أدى الى تعديل في شكل هذه التعليمات المتبنية من قبل هيئة الـ

(ANSI) وظهّرت تعليمية ACCEPT لتلائم هذا التطور فكان شكلها العام على النحو

التالي :

$$\begin{aligned}
 & \text{ACCEPT identifier - 1} \left[\text{, UNIT} \left\{ \begin{array}{l} \text{identifier - 2} \\ \text{literal - 1} \end{array} \right\} \right] \left[\text{, LINE} \right. \\
 & \left. \left\{ \begin{array}{l} \text{identifier - 3} \\ \text{literal - 2} \end{array} \right\} \right] \left[\text{, POSITION} \left\{ \begin{array}{l} \text{identifier - 4} \\ \text{literal - 3} \end{array} \right\} \right] \\
 & \left[\text{, SIZE} \left\{ \begin{array}{l} \text{identifier - 5} \\ \text{literal - 4} \end{array} \right\} \right] \left[\text{, PROMPT} \left[\text{literal - 5} \right] \right] \\
 & \left[\text{, ECHO} \right] \left[\text{, CONVERT} \right] \left[\text{, TAB} \right] \left[\text{, NO BEEP} \right] \\
 & \left[\text{, ERASE} \right] \left[\text{, OFF} \right] \left[\text{, LOW} \right] \\
 & \left[\text{, HIGH} \right] \\
 & \left[\text{, identifier - 6} \right] \left[\text{, UNIT} \left\{ \begin{array}{l} \text{identifier - 7} \\ \text{literal - 6} \end{array} \right\} \right] \left[\text{, LINE} \right. \\
 & \left. \left\{ \begin{array}{l} \text{identifier - 8} \\ \text{literal - 7} \end{array} \right\} \right] \left[\text{, POSITION} \left\{ \begin{array}{l} \text{identifier - 9} \\ \text{literal - 8} \end{array} \right\} \right] \\
 & \left[\text{, SIZE} \left\{ \begin{array}{l} \text{identifier - 10} \\ \text{literal - 9} \end{array} \right\} \right] \left[\text{, PROMPT} \left[\text{literal - 10} \right] \right]
 \end{aligned}$$

(*) T1990 COBOL

[, ECHO] [, CONVERT] [, TAB] [, NO BEEP]
 [, ERASE] [, OFF] [{ , LOW }] [{ , HIGH }]
 [ON EXCEPTION identifier - 11, imperative statement]

وتعليلة ACCEPT هذه تستخدم لنقل البيانات من خلال الشاشة الضوئية الى الحاسب لتتبع استخدامها ومعالجتها بواسطة البرنامج وتخزن هذه البيانات في المتغير المسمى بالحقل 1 - identifier و 6 - identifier ... وغيره .

وعبارة UNIT تعني تحديد رقم الشاشة التي يتم بموجبها إدخال المعلومات وقبولها الى داخل البرنامج ، وإذا لم تحدد فتكون هي الشاشة الذي يُشغل من خلالها البرنامج .

أما عبارات LINE و POSITION و SIZE فهي تحدد رقم السطر والموضع وحجم البيانات المدخلة في تعليمة القبول وينطبق عليها ما ورد في شرح تعليمة DISPLAY .

أما عبارة PROMPT فتفيد في تحديد حجم البيانات المراد قبولها وهي تُظهر، عند استخدامها، على الشاشة إشارة (*) أو (-) على عدد من المرات يساوي الحقل الموصف في قسم البيانات والمخصص لاستلام البيانات المطلوبة . وعند إضافة الثابت 5 - literal أو 10 - literal بجانبها عندئذ يمكن تحديد شكل آخر غير إشارة (*) للظهور قبل إدخال البيانات ويجب أن يكون أو تكون هذه الإشارة أو الرمز غير رقمي .

عبارة ECHO عند استخدامها تتيح إعادة عرض البيانات المقبولة للتأكد من صحة استلامها على الوجه المطلوب .

أما إذا كانت البيانات المراد ادخالها رقمية وتحتوي على فاصلة عشرية عندئذ يجب استخدام عبارة CONVERT مع التعليمة من أجل تخصيص الأماكن العشرية

والصحيحة وتخزينها في الوضع الصحيح أثناء عملية قبول البيانات . وفي حال عدم ذكرها في التعليمية يعتبر الرقم المدخل كله عدداً صحيحاً .

كلمة : TAB تمكن مدخل البيانات من التأكد من البيانات التي يدخلها وتتيح له قراءتها ثم ضغط الزر (الملمس) الخاص RETURN أو NEW - LINE لقبول المعلومات المدخلة ، وعدم ذكر هذه الكلمة يعني قبول البيانات المدخلة آلياً دون ضغط ملمس RETURN في نهاية الإدخال للحقل المطلوب .

ووضع عبارة NO BEEP تسمح بإخفاء صوت BEEP الذي تظهره الشاشة عند إدخال أي رمز . وإغفال هذه العبارة يؤدي الى ظهور الصوت BEEP (بيب) .

أما كلمة ERASE فتقوم عند استخدامها بمسح وتنظيف الشاشة قبل عملية ادخال الحقل (البيانات) المطلوبة بتعليمية ACCEPT .

أما كلمة OFF تسمح بادخال البيانات على الشاشة مع عدم ظهورها أثناء إدخالها من لوحة المفاتيح والملمس وتستخدم هذه الكلمة لادخال كلمات السر (PASS WORD) في البرنامج .

أما كلمتا LOW و HIGH فهي تستخدم كما ورد في تعليمية DISPLAY . أما وجود عبارة ON EXCEPTION مع تعليمية ACCEPT يتيح للمبرمج استخدام المفاتيح الخاصة المسماة FUNCTION KEYS وتختلف هذه عدداً وكيفاً باختلاف الحاسب المستخدم . إلا أن الشكل العام لهذه المفاتيح هو استخدامها كأداة مقارنة وشرط لقبول عمل ما أو الانتقال الى عمل آخر .

أمثلة :

- 1) ACCEPT FUNC - 1, FUNC - 2.
- 2) ACCEPT ANSWER - 1 LINE 1 POSITION MI PROMPT

ECHO CONVERT.

- 3) ACCEPT NEXT - L POSITION 0 PROMPT ECHO.
- 4) ACCEPT YEAR LINE 10 POSITION 10 MONTH LINE 11
POSITION 10.
- 5) ACCEPT ANSWER - 2 ON EXCEPTION STATUS - WD, GO
TO DSP - WRD.

٤ - ١٠ تعليمية «أفحص» INSPECT

تستخدم تعليمية «أفحص» INSPECT لأجل تدقيق أي حقل واستبدال رمز معين برمز آخر في هذا الحقل. وتفيد في شكل خاص أثناء عملية التحقق من البيانات. ويمكن أن تقوم هذه التعليمية بعد (احصاء) رمز معين ضمن الحقل ومعرفة عدد تكراره. ويمكن تلخيص التطبيقات الخاصة بهذه التعليمية في نقطتين :

- ١ - عد أو احصاء عدد مرات تكرار رمز معين Character في حقل ما identifier.
- ٢ - استبدال رمز معين ومكرر ضمن حقل ما برمز آخر.

ولهذه التعليمية شكلان عامان :

الشكل الأول :

INSPECT identifier - 1 TALLYING

$$\left\{ \text{identifier - 2 FOR } \left\{ \begin{array}{c} \underline{\text{ALL}} \\ \underline{\text{LEADING}} \\ \underline{\text{CHARACTERS}} \end{array} \right\} \left(\begin{array}{c} \text{identifier - 3} \\ \text{literal - 1} \end{array} \right) \right\}$$

$$\left[\left\{ \begin{array}{c} \text{BEFORE} \\ \text{AFTER} \end{array} \right\} \underline{\text{INITIAL}} \left\{ \begin{array}{c} \text{identifier - 4} \\ \text{literal - 2} \end{array} \right\} \right]$$

أمثلة :

- 1) INSPECT FIELD - 1 TALLYING CNTR - 1 FOR ALL SPACES.
- 2) INSPECT FIELD - 2 TALLYING CNTR - 2 FOR CHARACTERS BEFORE INITIAL SPACE.
- 3) INSPECT FLD - 3 TALLYING CNTR - 3 FOR LEADING ZEROS.
- 4) INSPECT FLD - 4 TALLYING CNTR - 4 FOR ALL "A" AFTER INITIAL "C".

يكون محتوى العدادات بعد التنفيذ : إذا كانت قيمة الحقل Identifier Value :

| | |
|-----------------------|--------------|
| FLD - 1 = \□□□ 51 □ | CNTR - 1 = 4 |
| FLD - 2 = 1725 □ | CNTR - 2 = 5 |
| FLD - 3 = 76 | CNTR - 3 = 2 |
| FLD - 4 = A B X C A A | CNTR - 4 = 2 |

وفي هذه الأمثلة فإن الحقول FLD - 1 الى FLD - 4 هي اما كن معرفة في قسم البيانات بوساطة عبارة PIC. وكذلك الحقول CNTR - 1 الى CNTR - 4 هي اما كن موصوفة في نفس القسم بوساطة عبارة PIC لأجل حفظ نتيجة عدد الرموز المطلوب (فحصها) أو عدها بوساطة التعليمة INSPECT TALLYING.

و يقصد أيضا بكلمة ALL جميع الرموز وكلمة LEADING الرموز الموجودة في مقدمة الحقل من جهة اليسار.

الشكل الثاني

وتعليمة «أفحص» يمكن أن تقوم بوظيفتها الثانية وهي استبدال الرموز في الحقل عند إضافة جملة REPLACING بدلاً من جملة TALLYING و يصبح شكلها كما يلي :

INSPECT identifier - 1 REPLACING

$$\left\{ \begin{array}{l} \underline{\text{ALL}} \\ \underline{\text{LEADING}} \\ \underline{\text{FIRST}} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier - 2} \\ \text{literal - 1} \end{array} \right\}$$

CHARACTERS

$$\underline{\text{BY}} \left\{ \begin{array}{l} \text{identifier - 3} \\ \text{literal - 2} \end{array} \right\} \left[\begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right] \text{INITIAL} \left\{ \begin{array}{l} \text{identifier - 4} \\ \text{literal - 3} \end{array} \right\}$$

حيث أن معنى كلمة ALL و LEADING يأخذ نفس المعنى في الشكل الأول وكلمة FIRST يقصد بها أول رمز. وهذه التعليلة تقوم باستبدال الرمز المعروف في الحقل identifier - 2 و literal - 1 بالرمز المعروف في الحقل identifier - 3 أو الثابت literal - 2.

أمثلة :

- 1) INSPECT FLD - E REPLACING ALL "1" BY "2".
- 2) INSPECT FLD - E REPLACING LEADING "1" BY "2".
- 3) INSPECT FLD - E REPLACING CHARACTERS BY "3" BEFORE INITIAL "2".
- 4) INSPECT FLD - E REPLACING FIRST "X" BY "Y".

نكون قيمة الحقل FLD-E بعد التنفيذ : إذا كانت قيمة الحقل FLD-E قبل التنفيذ :

- | | |
|-------------|----------|
| 1) = 112111 | = 222222 |
| 2) = 112111 | = 222111 |

مثال :

MOVE ZERO TO CNT - X.

INSPECT FLD - 1 TALLYING CNT - X FOR LEADING "B"

REPLACING ALL "B" BY "R".

فإذا كان الحقل 1 - FLD في المثال السابق يحتوي على البيانات التالية قبل تنفيذ
تعليلة INSPECT :

FLD - 1 = BBBBABC

وكذلك يكون CNT - X مساوياً للصفر بواسطة تعليلة MOVE فإن هذين الحقلين
يصبحان بعد التنفيذ على النحو التالي :

FLD - 1 = RRRRARCD

CNT - X = 04

حيث أنهما يأخذان الصورة التالية في التوصيف في قسم البيانات :

FLD - 1 PIC X (8).

CNT - X PIC 99.

وتجدر الإشارة هنا إلى أن تعليلة INSPECT التي أدخلت إلى لغة كوبول في
التحسينات الأخيرة (عام ١٩٧٤) تعادل إلى حد قريب تعليلة EXAMINE التي
كانت موجودة في نسخة لغة كوبول قبل عام ١٩٧٤م وكانت تعليلة EXAMINE
تأخذ الشكلين التاليين :

| | | |
|---------------------------------------|--|-------------|
| 1) <u>EXAMINE</u> identifier TALLYING | $\left\{ \begin{array}{l} \underline{\text{UNTIL FIRST}} \\ \underline{\text{ALL}} \\ \underline{\text{LEADING}} \end{array} \right\}$ | literal - 1 |
| [REPLACING BY literal - 2] | | |

-
- STRUCTURED COBOL PROGRAMMING, stern / stern, John Wiley , U.S.A, 1980, 3ed. P.P. 364

حيث يتم تخزين عدد الرموز المحدودة (المحصاة) في الحقل TALLY الذي يقوم نظام التشغيل (أو مترجم الكوبول) بتعيينه وتعريفه ويستطيع المبرمج معالجته بهذا الاسم. أما الشكل الثاني المخصص لاستبدال الرموز فهو على النحو التالي :

2) EXAMINE identifier REPLACING

| | |
|--|-----------------------------------|
| $\left\{ \begin{array}{l} \underline{\text{ALL}} \\ \underline{\text{LEADING}} \\ \underline{\text{FIRST}} \\ \underline{\text{UNTIL FIRST}} \end{array} \right\}$ | literal - 1 <u>BY</u> literal - 2 |
|--|-----------------------------------|

مثال :

EXAMINE NAME1 REPLACING ALL "0" BY SPACE.

٤ - ١١ أسئلة وتمارين

سؤال ١ - بين الصحيح من العبارات التالية بإشارة (✓) والخطأ منها بإشارة (x):

- ☐ ١ - تستخدم عبارة التطابق CORRESPONDING مع تعليمة النقل MOVE فقط.
- ☐ ٢ - يمكن استخدام عبارة ROUNDED وعبارة ON SIZE ERROR مع أي من التعليمات الحسابية.
- ☐ ٣ - يجب أن تحتوي تعليمة COMPUTE عبارة ROUNDED لإعطاء القيمة المحسوبة المقربة.

- ٤ - عند استخدام تعليمة ACCEPT للحصول على اليوم DAY يمكن معرفة اسم اليوم في السنة مباشرة .
- ٥ - عند استخدام تعليمة ACCEPT للحصول على الوقت TIME يتطلب حجز حقل طوله ستة خانات (digits) للحصول عليه .
- ٦ - استخدام تعليمة WRITE مع تعليمة FROM يعني عن استعمال تعليمة MOVE قبلها لأجل نقل البيانات المراد كتابتها .
- ٧ - تأخذ تعليمة INSPECT شكلين مختلفين .

سؤال ٢ - إذا كانت لدينا تعليمة IF المتداخلة على النحو التالي :
الشكل الأول

```
IF M-STATUS = 'S'
    PERFORM SINGLE - RTN
ELSE
    IF M-STATUS = 'M'
        PERFORM MARRIED - RTN
    ELSE
        PERFORM ERROR - RTN.
```

وكان لدينا أيضا عدة تعليمات IF وفق الشكل التالي :
الشكل الثاني

```
IF M-STATUS = 'M' PERFORM MARRIED - RTN.
IF M-STATUS = 'S' PERFORM SINGLE - RTN.
```

IF M-STATUS NOT = 'S'

AND M-STATUS NOT 'M' PERFORM ERROR - RTN.

والمطلوب إيضاح ما يلي :

- ١ - هل الشكل الأول يعادل الشكل الثاني و يعطي نفس النتيجة.
 - ٢ - في حال كون هذين الشكلين متكافئين فايهما أفضل من ناحية البرمجة وحسن تنفيذ البرنامج .
 - ٣ - هل تكون نتيجة التنفيذ واحدة لو استبدلنا العامل AND بالعامل OR في الشكل الثاني .
- سؤال ٣ - لدينا الجزء التالي من برنامج كوبرول والذي وضع لحساب عمر الفرد الموظف من واقع سجله RECORD المخزن على وسيط مغناطيسي ، ويحتوي هذا السجل على تاريخ ميلاده EMP - BRITH - DATE ، أما تاريخ التنفيذ للبرنامج فيتم الحصول عليه بواسطة تعليمة ACCEPT مع عبارة DATE لأجل حساب الفترة الزمنية السابقة ومعرفة العمر الصحيح .
- وعلى الرغم من أن ترميز هذا البرنامج صحيح من الناحية الشكلية إلا أنه لا يعطي النتيجة المطلوبة عند تنفيذه لأجل معرفة العمر . فالمطلوب معرفة أسباب ذلك وتصحيح هذا البرنامج على ضوء قواعد استخدام تعليمة ACCEPT وطريق حساب العمر .

١ EMP - REC.

٢ EMP - BIRTH - DATE.

٣ BIRTH - MONTH PIC 99.

٤ BIRTH - YEAR PIC 99.


```

01  EMPL - AGE          PIC 99.
    -----
01  DATE - WORK - AREA.
    05  TODAYS - MONTH   PIC 99.
    05  TODAYS DAY       PIC 99.
    05  TODAYS - YEAR    PIC 99.
    -----
PROCEDURE DIVISION.

    ACCEPT DATE - WORK - AREA FROM DATE.
    -----
    COMPUTE EMPL - AGE = TODAYS - YEAR
        - BIRTH - YEAR + TODAYS - MONTH
        - BIRTH - MONTH.

```

سؤال ٤ - إذا كان لدينا حقل اسمه CITY - CODE وطوله رمز واحد (بايت) والممكن أن يحتوي هذا الحقل على أحد الرموز التالية :

S, M, D, W

والمطلوب توصيف هذا الحقل مستخدماً اسم الشرط Condition - name في قسم البيانات .

تمرين ١ :

أكتب برنامجاً بلغة الكوبول يقبل عن طريق الشاشة TERMINAL رقم الموظف، واسمه، وراتبه وبذل النقل، ويسجل هذه المعلومات على ملف متسلسل SERIAL

موجود على القرص الممغنط واسمه SALFILE وشكل السجل فيه كالتالي :

| | FIELD NAME | PICTURE |
|------------|------------|------------|
| رقم الموظف | EM - NO | 9 (4) |
| اسمه | EM - NAME | X (20) |
| راتبه | EM - SAL | 9 (5) V 99 |
| بدل النقل | EM - TRANS | 9 (3) V 99 |
| غير مستعمل | BLANK | X (4) |

أظهر الرسائل المناسبة عند إدخال كل حقل .

وفي نهاية العمل أدخل الرمز المناسب للانتهاء .

• أعمل تحقق للحقول السابقة بحيث أن :

رقم الموظف وراتبه وبدل النقل يجب أن تكون رقمية وأكبر من الصفر أما اسم الموظف فيجب ألا يكون فراغاً .

تقرين ٢ :

احدى الشركات الخاصة تريد عمل صندوق لمساعدة العمال ، وذلك باستقطاع مبلغ معين من الأجر الشهري لكل عامل وفق القواعد التالية :

أ. اذا كان أجر العامل ١٠٠٠ ريال وما دون يقتطع عليه ١ % من الأجر الشهري لصالح الصندوق .

ب. اذا كان أجر العامل أكبر من ١٠٠٠ ريال وأقل من ٢٠٠١ ريال يقتطع عليه :

١ % عن الألف الأولى من أجره الشهري

٣ % عن المبلغ المتبقي من الأجر الشهري

جـ. إذا كان أجر العام أكبر من ٢٠٠٠ ريال يقطع عليه :

١% عن الألف الأولى من الأجر الشهري

٣% عن الألف الثانية من الأجر الشهري

٥% عن باقي الأجر الشهري (ما فوق ٢٠٠٠ ريال)

المطلوب :

اكتب برنامجاً يحسب صافي الأجر لكل عامل علماً بأنه يحسم عليه ٥% من الأجر الشهري (قبل حسم الصندوق) كتأمينات اجتماعية.

علماً بأن الملف موجود على قرص ممغنط اسمه PAYFILE وأن توصيف السجل فيه كالتالي :

| | FIELD NAME | PICTURE |
|--------------|-------------|------------|
| رقم الموظف | EMP - NO | 9 (4) |
| اسمه | EMP - NAME | X (24) |
| الأجر الشهري | EMP - SAL | 9 (5) V 99 |
| بدل النقل | EMP - TRANS | 9 (3) V 99 |
| | BLANK | X (4) |

واطبع رقم الموظف، اسم الموظف، الأجر الشهري، بدل النقل، تبرع الصندوق، التأمينات الاجتماعية، صافي الراتب، على الطابعة من أجل كل عامل - أنظر شكل المخرجات صفحة ٦٠٢ والنتائج صفحة ٥٩٩.

١ - اطبع رقم الصفحة والتاريخ مع العنوان في رأس كل صفحة

٢ - اترك سطر فراغ بين كل سطري طباعة

٣ - لا تطبع في الصفحة الواحدة أكثر من ٢٥ سطراً

أمثلة على طريقة حساب صافي الأجر الشهري :

مثال ١ - بفرض أن أحد العمل أجره الشهري ٢٥٠٠ ريال وبدل النقل ٣٠٠ ريال
فيكون

١ - التأمينات

$$\text{ريال } ١٢٥ = \frac{٥ \times ٢٥٠٠}{١٠٠}$$

٢ - التبرع

$$\text{ريالات } ١٠ = ١\% \times ١٠٠٠$$

$$\text{ريال } ٣٠ = ٣\% \times ١٠٠٠$$

$$\text{ريال } ٢٥ = ٥\% \times ٥٠٠$$

المجموع
ريال ٦٥

$$\text{٣ - الصافي} = (٢٥٠٠ + ٣٠٠) - (١٢٥ + ٦٥) = ٢٦١٠$$

مثال ٢ - الأجر الشهري ١٢٠٠ ريال وبدل النقل ٣٠٠ ريال يكون :

$$\text{١ - التأمينات} = \frac{٥ \times ١٢٠٠}{١٠٠} = \text{ريال } ٦٠$$

$$\text{٢ - التبرع} = ١\% \times ١٠٠٠ = \text{ريالات } ١٠$$

$$\text{ريالات } ٦ = ٣\% \times ٢٠٠$$

المجموع
ريال ١٦

$$\text{٣ - الصافي} : (١٢٠٠ + ٣٠٠) - (٦٠ + ١٦) = ١٤٢٤ \text{ ريال}$$

تمرين ٣ - ملف من البطاقات المثقبة اسمه STDFILE يحتوي على اسم الدارس ودرجته النهائية بمعدل بطاقة لكل دارس :

| | FIELD NAME | PICTURE |
|-----------------|------------|---------|
| اسم الدارس | STD - NAME | X (25) |
| الدرجة النهائية | STD - MARK | 9 (3) |
| _____ | FILLER | X (52) |

اكتب برنامجاً بلغة الكوبول يقرأ هذا الملف ويطبع محتواه على الطابعة مع ملاحظة القواعد التالية :

- اذا كانت الدرجة STD- MARK أقل من ٦٠ أطبع كلمة 'FAIL' راسب أمام الدرجة

- اذا كانت الدرجة من ٦٠ إلى ٦٩ أطبع كلمة 'PASS' ناجح

- اذا كانت الدرجة من ٧٠ إلى ٧٩ أطبع كلمة 'GOOD' جيد

- اذا كانت الدرجة من ٨٠ إلى ٨٩ أطبع كلمة 'V.GOOD' جيد جداً

- اذا كانت الدرجة تسعين فما فوق اطبع كلمة 'EXCELLENT' ممتاز

مع ملاحظة أن الدرجة العظمى هي ١٠٠ .

شكل سجل الاخراج :

| | | |
|----------------------|--------|----------|
| XXXXXXXXXXXXXXXXXXXX | 999 | XXXXXXXX |
| للاسم | للدرجة | للتقدير |

الفصل الخامس

أقسام برنامج كوبول الأخرى والعبارات المتقدمة

٥ - ١ لمحة عامة :

يفيد هذا الفصل بالإضافة الى المبتدئين في برمجة كوبول، المبرمجين الذين يودون التقدم أكثر في لغة كوبول حيث أنه يزود القارئ بتفاصيل كثيرة عن بقية أقسام البرنامج (قسم البيانات) و (قسم المحيط). وخاصة فيما يتعلق باظهار أو اخراج البيانات والرموز المستخدمة لذلك، وبقية توصيف البيانات مثل عبارة USAGE وأسماء الشرط وعبارات إعادة التوصيف والتسمية REDEFINES, RENAMES علاوة على ذكر تعليمة النسخ COPY، وفي نهاية الفصل ترد البرامج الفرعية والرئيسية وكيفية الربط بينهما وفصل الربط LINKAGE SECTION .

٥ - ٢ رموز الاظهار (الاخراج) * EDITING CHARACTERS

تعتبر الرموز التالية من الرموز المساعدة على إخراج الحقل (عرض البيانات) على

شكل معين : B, O, +, -, Z, @, \$, ., /, CR, DB

وهذه الرموز تشير الى عملية الاظهار المطلوبة في الحقل . ويمكن تصنيف هذه الرموز الى أربعة أصناف :

١ - عدم إظهار بسيط (Simple Supression) : وهي الرموز التي تسبب عدم ظهور

(*) Sordillo Donald, THE PROGRAMMERS ANSI COBOL REFERENCE MANUAL,

Prentice - Hall, 1978, PP, 193-203.

بعض البيانات في الحقل .

- ٢ - عدم إظهار واستبدال Suppression with Replacement : وهي الرموز التي تسبب عدم ظهور بعض البيانات وبنفس الوقت تمكن من استبدالها بشكل آخر.
- ٣ - الاستبدال الفاتض Floating Replacement : وهي الرموز التي تسبب إدخال رمز أو رموز في الحقل عند إظهار بياناته في حالة معينة .
- ٤ - الإدخال (الاقحام) الثابت Fixed Insertion : وهي الرموز التي تؤدي الى إدخال رمز أو رموز وتبقى ثابتة عند إظهار البيانات .

٥ - ٢ - ١ الطمس البسيط Simple Suppression

يؤدي الرمز Z الى طمس جميع الأصفار التي تظهر في الحقل الرقمي من جهة اليسار (أي الأصفار غير المعنوية) وكذلك يمكن طمس قيمة الحقل المراد إظهاره اذا كانت قيمته صفراً وكان حقل الاظهار Edited field موصوفا بالرمز Z .

واذا كان الحقل المراد إظهاره يحتوي على رقم عشري وتم وصف الجزء الصحيح منه بالرمز Z فإن الأصفار التي تقع على جهة اليسار من الجزء الصحيح فقط تطمس . أما اذا كان الجزء العشري من الحقل موصوفاً بالرمز Z فيجب أن تكون جميع الأرقام التالية من جهة اليسار موصوفة به كذلك . ولا يتم طمس الأصفار في هذه الحالة إلا على يسار الفاصلة العشرية . واذا كان الرقم بكامله مساوياً للصفر فإن القيمة المظهرة تكون فراغاً حتى بدون ظهور الفاصلة العشرية . لأن أي رمز ثابت Fixed insertion مقحم في شكل الحقل يعامل مثل الرمز Z .

والجدول التالي * يبين الأشكال المختلفة لاطهار حقل رقمي :

(a) Sordillo P. 195.

| Data | PICTURE | | | | | | |
|---------|---------|------|------|------|------|-------|-------|
| | 9999 | Z999 | ZZ99 | ZZZ9 | ZZZZ | ZZ.ZZ | Z.ZZZ |
| 0 | 0000 | 000 | 00 | 0 | | | |
| 1.01 | 0001 | 001 | 01 | 1 | 1 | 1.01 | 1 |
| 217. | 0217 | 217 | 217 | 217 | 217 | 17.00 | 217 |
| 1010.20 | 1010 | 1010 | 1010 | 1010 | 1010 | 10.20 | 1,010 |
| .01 | 0000 | 000 | 00 | 0 | | .01 | |

٥-٢-٢ الطمس مع الاستبدال

SUPPRESSION WITH REPLACEMENT

يتم طمس الأصفار من جهة يسار القيمة مع استبدالها بإشارة النجمة * بدلاً من الفراغ بواسطة الرمز المستخدم للإظهار «*» والمسمى Asterisk. وعند استخدام النجمة كرمز لتوصيف حقل الإظهار يضاف إلى جانب الحقل الرقمي المراد إظهاره من جهة اليسار نجوماً بدلاً من الأصفار غير المعنوية. وتستخدم هذه الطريقة في إظهار الحقول الرقمية بشكل خاص في طباعة الشيكات المصرفية أو تدوين القيمة المالية تلافياً لامكانية زيادة القيمة والتزوير.

ويجب توصيف الحقل بعدة نجوم متصلة قد تفصل بينها رموز ثابتة

(Fixed Insertion Characters).

ولا يمكن استخدام رمز النجمة اذا كان الحقل المراد إظهاره موصوفاً بعبارة

. BLANK WHEN ZERO

الجدول التالي يوضح كيفية استخدام توصيف رمز النجمة في إظهار الحقول الرقمية :

| PICTURE | | | | | |
|---------|------|------|------|-------|----------------------|
| Data | 999 | 99 | 9 | 00 | 00.00 |
| 0 | 000 | 00 | 0 | 00 | 0000.00 ¹ |
| 13 | 013 | 013 | 13 | 13.00 | 13.00 |
| 123 | 123 | 123 | 123 | 23.00 | 0123.00 |
| 2010 | 2010 | 2010 | 2010 | 10.00 | 2010.00 |
| 1010.2 | 1010 | 1010 | 1010 | 10.20 | 1010.20 |
| .01 | 000 | 00 | 0 | 00.01 | 0000.01 ¹ |

٥ - ٢ - ٣ الاستبدال العائم FLOATING REPLACEMENT

يستخدم لأجل الاستبدال العائم في إظهار الحقول ثلاثة رموز وهي :

Currency Symbol علامة العملة المتداولة

+ إشارة الموجب

- إشارة السالب

وتُمثل علامة النقد برمز واحد مثل (\$) أو (£) أو أي رمز آخر حسب نوع العملة (على أن يكون هذا الرمز متوفر على وحدات الإدخال والإخراج) . ويتم تحديده في عبارة CURRENCY SIGN IS في الفقرة المسماة للأسماء والتعريفات الخاصة ؛ من قسم المحيطات SPECIAL - NAMES .

(٥) Sordillo P. 196

(١) يلاحظ أن مكان الفاصلة العادية ظهر نجمة أما الفاصلة البشرية فقد ظهرت كما هي.

كما يمكن أن تكون علامة العملة المتداولة رمز اقحام ثابت . وعند استخدامها كرمز عائم تظهر العلامة الى جانب آخر رقم من جهة اليسار ويكون الجزء التالي لها من الحقل فراغاً (أيضاً من جهة اليسار).

والجدول التالي • يوضح كيفية استخدام هذا الرمز في الإظهار :

| Data | PICTURE | | | |
|---------|----------|---------|------------|------------------|
| | \$\$\$99 | \$\$\$9 | \$\$\$9.99 | \$\$,\$\$\$.\$\$ |
| 0 | \$00 | \$0 | \$00 | |
| .02 | \$00 | \$0 | \$02 | \$02 |
| 12 | \$12 | \$12 | \$12.00 | \$12.00 |
| 12.34 | \$12 | \$12 | \$12.34 | \$12.34 |
| 1234 | \$234 | \$1234 | \$1234.00 | \$1,234.00 |
| 1234.56 | \$234 | \$1234 | \$1234.56 | \$1,234.56 |

أما بالنسبة للاشارتين الجبريتين السالب والموجب فإنهما يمكن أن يستخدموا إما كرموز ثابتة أو عائمة . وإذا استخدم الرمز «علامة السالب» فإن إشارة السالب سوف تظهر في الحقل عند كون قيمة الحقل الأصلي سالبة أما اذا كانت القيمة موجبة أو صفراً فإن الإشارة لا تظهر. أما إذا استخدمت الإشارة الموجبة كرمز فإن الإشارة الجبرية للقيمة سوف تظهر سواء كانت القيمة سالبة أم موجبة .

والشكل التالي •• يوضح كيفية استخدام هذين الرمزتين في إظهار البيانات الرقمية .

(•) Sordillo P. 196

(••) Sordillo P. 197

| Data | PICTURE | | | | | | | |
|--------|---------|-------|-------|-------|-------|-------|----------|-----------|
| | +9999 | -9999 | +++9 | ---9 | ++++ | ---- | ++++.++ | ---.--- |
| 0 | +0000 | 0000 | +0 | 0 | | | | |
| 12 | +0012 | 0012 | +12 | 12 | +12 | 12 | +12.00 | 12.00 |
| -12 | -0012 | -0012 | -12 | -12 | -12 | -12 | -12.00 | -12.00 |
| 123 | +0123 | 0123 | +123 | 123 | +123 | 123 | +123.00 | 123.00 |
| -123 | -0123 | -0123 | -123 | -123 | -123 | -123 | -123.00 | -123.00 |
| -123 | -0123 | -0123 | -123 | -123 | -123 | -123 | -123.00 | -123.00 |
| 1234 | +1234 | 1234 | +1234 | 1234 | +1234 | 1234 | +1234.00 | 1,234.00 |
| -1234 | -1234 | -1234 | -1234 | -1234 | -1234 | -1234 | -1234.00 | -1,234.00 |
| 12.34 | +0012 | 0012 | +12 | 12 | +12 | 12 | +12.34 | 12.34 |
| -12.34 | -0012 | -0012 | -12 | -12 | -12 | -12 | -12.34 | -12.34 |

٥-٢. ٤ الاقحام الثابت FIXED INSERTION

إن الرموز المستخدمة في الاقحام الثابت عند اظهار حقل ما هي : الفاصلة ، الصفر 0 و / و B (تعني فراغ Blank) و CR (تعني دائن Credit) و DB (تعني مدين Debit) . وكذلك وجود رمز واحد فقط في توصيف حقل الاظهار من الرموز التالية - و + واسارة العملة المتداولة (\$) .

تُحصى جميع الرموز الثابتة عند حساب حجم الحقل المراد اظهاره وإشارتا CR و DB تأخذ موضعين عند تحديد حجم الحقل . كما يجب الاشارة الى أن الفاصلة (وليس المقصود الفاصلة العشرية بل الفاصلة التي تحدد الألوف في الرقم) لا يمكن أن تكون آخر رمز في توصيف الحقل المُظهر.

والشكل التالي • يوضح الاستخدامات المختلفة لهذه الرموز في اظهار حقن البيانات :

| Data | PICTURE | | | | | |
|------|---------|-------|--------|--------|---------|-------|
| | 99B99 | 9,999 | 99/909 | ••,••• | 9B9/900 | 99,99 |
| 0 | 00 00 | 0,000 | 00/000 | •••••• | 0 0/000 | 00,00 |
| 12 | 00 12 | 0,012 | 00/102 | ••••12 | 0 1/200 | 00,12 |
| 123 | 01 23 | 0,123 | 01/203 | •••123 | 1 2/300 | 01,23 |
| 1234 | 12 34 | 1,234 | 12/304 | •1,234 | 2 3/400 | 12,34 |

أما إشارتا الدائن والمدين DB و CR فيجب وضعها على يمين الرمز الموصفة للحقل المظهر. ولا يمكن استخدامها مع الحقول الموصوفة بالرموز التالية : A - X + أو S كما لا يمكن وضع الاشارتين في نفس التوصيف الواحد. ومن الشكل التالي •• يتبين أن إشارة الدائن أو المدين تظهر في الحقل عند وجود القيمة سالبة فقط. وتختمفي عند كون القيمة صفر أو موجبة .

| Data | PICTURE | | | | | |
|-------|----------|----------|---------------|---------------|--------|--------|
| | \$9999CR | \$9999DB | \$\$,\$\$\$CR | \$\$,\$\$\$DB | ZZZZCR | ZZZZDB |
| 0 | \$0000 | \$0000 | | | | |
| 12 | \$0012 | \$0012 | \$12 | \$12 | 12 | 12 |
| -12 | \$0012CR | \$0012DB | \$12CR | \$12DB | 12CR | 12DB |
| 1234 | \$1234 | \$1234 | \$1,234 | \$1,234 | 1234 | 1234 |
| -1234 | \$1234CR | \$1234DB | \$1,234CR | \$1,234DB | 1234CR | 1234DB |

(•) Sordillo 198

(••) Sordillo 198

هذا ويمكن توليف أكثر من رمز في توصيف الحقل المظهر. كما يمكن وضع
الاشارتين الجبريتين على يمين الحقل الموصوف بدلاً من جهة الشمال والجدول التالي
يعطي ملخصاً عن كيفية هذه الاستخدامات المختلفة.

| Data | PICTURE | | | | | | | |
|------|---------|------|------|------|--------|--------|----------|----------|
| | +999 | -999 | 999+ | 999- | +\$999 | -\$999 | \$####CR | \$ZZZZCR |
| 0 | +000 | 000 | 000+ | 000 | +\$000 | \$000 | \$#### | |
| 12 | +012 | 012 | 012+ | 012 | +\$012 | \$012 | \$#012 | \$ 12 |
| -12 | -012 | -012 | 012- | 012- | -\$012 | -\$012 | \$#012CR | \$ 12CR |
| 123 | +123 | 123 | 123+ | 123 | +\$123 | \$123 | \$#123 | \$ 123 |
| -123 | -123 | -123 | 123- | 123- | -\$123 | -\$123 | \$#123CR | \$123CR |

٥-٢-٥ الفاصلة العشرية DECIMAL POINT

لا يمكن تصنيف الفاصلة العشرية ضمن أي نوع من أنواع الرموز السابقة لأنها لا
تعتبر فقط رمز اقحام ثابت Fixed insertion بل تدل على الجزء العشري والجزء الصحيح
من العدد. كما لا يمكن أن يظهر في الحقل إلا فاصلة عشرية واحدة (يعبر عنها باللغة
الانكليزية Point أي النقطة) وكذلك لا يمكن أن تُستخدم مع وجود الرموز التالية في
نفس التوصيف للحقل A و X و P و V. وهذه الفاصلة تقوم بتحديد الجزء العشري
والصحيح من الحقل الرقمي الحقيقي (Real) ووظيفتها مثل وظيفة الرمز V أو P. وعند
الظهور تُعد الفاصلة مع حساب حجم الحقل المظهر ولا يمكن أن تكون آخر رمز في رموز
توصيف الحقل.

(٥) Sordillo P. 199

وتجب الإشارة الى أن عبارة DECIMAL - POINT IS COMMA التي تكتب في
فقرة الأسماء الخاصة SPECIAL - NAMES في قسم المحيطيات تمكن المبرمج من
استبدال (النقطة العشرية Decimal Point) بفاصلة عشرية تحت نفس الشروط .

٥ - ٣ عبارة إعادة التوصيف REDEFINES

من أجل توفير المساحة المعروفة في قسم البيانات وتوفير المرونة في استخدام نفس
المكان الموصف في الذاكرة بأكثر من شكل وجدت عبارة REDEFINES والتي تعني
إعادة تعريف أو توصيف نفس الحقل (أو المتغير) باسم آخر أو شكل (PICTURE)
آخر.

ويمكن أن تأخذ عبارة إعادة التوصيف أي رقم مستوى Level No. من 01 الى 49
وتأتي بعد الحقل المراد توصيفه مباشرة دون أي فاصل أو انقطاع .

Level No. data - name - 1 PICTURE clause

Level No. data - name - 2 REDEFINES data - name - 1 [PIC clause]

(هـ) الرمز P يستخدم في توصيف الحقل للدلالة على وجود فاصلة عشرية مفترضة في العدد ولا يمكن أن يأتي مع الرمز V ويمكن
أن يكون من جهة اليمين أو اليسار في توصيف الحقل الرقمي المراد اظهاره .

مثال :

الرقم 12345 لو كان موصوفاً بالشكل التالي

PICTURE

فانه يظهر على الشكل 0000012345

أما اذا كان موصوفاً بما يلي 99999PICTURE

فانه يظهر على الشكل 1234500000

حيث أن كل P يمثل صفراً عند إظهار الحقل ولا تدخل في حساب الحجم داخل الذاكرة .

مثال :

Ø1 REC - ONE.

Ø5 FLD - IN - A ' PIC X (5).

Ø5 ' FLD - IN - B PIC 99.

Ø1 REC - TWO REDEFINES REC - ONE PIC X (7).

أو :

Ø2 UNIT - PRICE PIC 9 (4).

Ø2 D - UNIT - PRICE REDEFINES UNIT - PRICE PIC 99V99.

ويجب التأكيد هنا على أن يكون طول الحقل المعاد توصيفه في نفس الحجم دون زيادة أو نقصان في عدد الرموز (البايت) المكونة له .

كما أن قواعد لغة كوبيول لا تسمح بإضافة عبارة تحديد القيمة VALUE مع عبارة REDEFINES ، لأن ذلك يتعارض منطقياً (اعطاء قيمتين مختلفتين لنفس الحقل) .

كما تحذر الإشارة هنا الى أنه يمكن إعادة توصيف الحقل الواحد أكثر من مرة . أي يمكن استعمال أكثر من عبارة REDEFINES لنفس الحقل .

5 - 4 توصيف الإشارة الجبرية مع العدد SIGNED NUMBERS

يضاف الحرف S على وصف الحقل الرقمي في بدايته من أجل الإشارة الى أن هذا الحقل يحتوي على عدد مع إشارة جبرية والحرف S ليس له أهمية اذا كان الرقم المخزن في الحقل موجبا ولكن أهميته تظهر إذا كان الحقل يحتوي على عدد سالب . أو كان نتيجة عملية حسابية جمع أو طرح . وإذا حذف الحرف S من التوصيف فان الحقل

يفترض أن يكون موجباً . ولتأخذ الحقول الموصفة فيما يلي :

| | | | | |
|----|-----------|-----|-----|----------------|
| φ5 | FIELD - A | PIC | S99 | VALUE IS - 3φ. |
| φ5 | FIELD - B | PIC | 99 | VALUE IS - 25. |

فإذا كانت نتيجة عملية حسابية سوف تخزن في الحقل FIELD - A فإن ذلك يضمن صحة النتيجة لوجود مكان للإشارة الجبرية . أما إذا كانت نتيجة عملية حسابية ما سوف تتم وتخزن نتيجتها على الحقل FIELD - B فإن العدد الناتج سوف لا يكون صحيحاً في حال كون القيمة الناجمة سالبة لأن الحقل لا يحتوي موضعاً للإشارة . ويُفترض عندئذ الرقم كعدد موجب . فمثلاً في حالة جمع العددين A و B ونخزن النتيجة في A فإن الرقم الناتج يكون 5- بينما إذا خزنت النتيجة في B فإن النتيجة سوف تكون 5 بدون إظهار الإشارة وهذا غير صحيح جبرياً وحسابياً .

٥ - ٤ - ١ عبارة الإشارة SIGN CLAUSE

إن عبارة SIGN اختيارية وعند وجودها بجانب الحقل الموصوف في قسم البيانات تعطي الدلالة على تخصيص BYTE بايت للإشارة الجبرية المصاحبة للرقم وتعطي أيضاً المبرمج تحديد مكان الإشارة بالنسبة للرقم (من جهة اليمين أو من جهة اليسار) وتستخدم هذه العبارة فقط عندما يكون الحقل الرقمي موصوفاً بعبارة DISPLAY مع كلمة USAGE ، وعدم ذكرها بجانب الحقل أي DISPLAY يعني أنها معرفة ما لم يُعرف بديل آخر.

والشكل العام لعبارة الإشارة كما يلي :

$$[\text{SIGN IS}] \left\{ \begin{array}{l} \text{LEADING} \\ \text{TRAILING} \end{array} \right\} [\text{SEPARATE CHARACTER}]$$

حيث أن ظهور كلمة SEPARATE يدل على تخصيص بايت للإشارة أما عدم وجودها فيعني أن الرقم يتضمن الدلالة على اشارته الموجبة والسالبة بدون بايت خاصة بها . أما كلمة LEADING فيقصد بها أن تُعرّف الإشارة قبل الرقم من جهة اليسار (أي على يسار الرقم مثل 915-) أما كلمة TRAILING فيقصد بها تعريف الإشارة على يمين الرقم أو في نهايته (مثل -405) .

مثال : ليكن الحقل NUMBER معرف على النحو التالي

01 NUMBER PIC S999.

وإذا كانت قيمته مساوية للعدد 365 بأن يأخذ الأشكال التالية مع استخدام عبارة SIGN

01 NUMBER PIC S999 SIGN IS TRAILING.

3 6 5

01 NUMBER PIC S999 SIGN IS LEADING

3 6 5

01 NUMBER PIC S999 SIGN IS TRAILING SEPARATE.

رمز الإشارة الموجبة 3 6 5

01 NUMBER PIC S999 SIGN IS LEADING SEPARATE.

رمز الإشارة الموجبة 3 6 5

أما إذا كانت قيمة الحقل NUMBER مساوية للعدد 365- فيمكن أن يأخذ الأشكال التالية :

... SIGN IS TRAILING. الرقم 5 مضافاً إليه
قيمة الإشارة السالبة 3 6 5

... SIGN IS LEADING.

الرقم 3 مضافاً اليه
قيمة الاشارة السالبة



.... SIGN IS TRAILING SEPARATE.

... SIGN IS LEADING SEPARATE.



وتجدر الإشارة هنا إلى أن رمز الاشارة أو القيمة المضافة على الرقم للدلالة عليها تختلف باختلاف الحاسب المستخدم.

وعدم ذكر عبارة الإشارة SIGN يعني أن الرقم يحتوي على قيمتها ولا تدخل في بايت خاصة بها.

كما يجب توصيف الرمز S في الصورة (PIC) عند استخدام عبارة SIGN للدلالة على أن الحقل الموصوف معرف برقم مع إشارة جبرية.

5 - 5 عبارة صيغة التوصيف USAGE

إن وضع البيانات في ذاكرة الحاسب وتخزينها على الوحدات المتصلة به مهم جداً لأنه يحدد طريقة المعالجة المرغوبة للبيانات و يرفع من كفاية الأداء في البرنامج وعبارة USAGE توصف الشكل الذي تكون عليه البيانات داخل الحاسب. ويمكن لهذه العبارة أن توصف الحقول الجزئية Elementary items أو الاجالية Group items وفي أي جزء من قسم البيانات DATA DIVISION.

والصيغة العامة لاستخدام عبارة USAGE تأخذ الشكل التالي :

$$\text{USAGE IS } \left\{ \begin{array}{l} \text{DISPLAY} \\ \left\{ \begin{array}{l} \text{COMPUTATIONAL - 3} \\ \text{COMP - 3} \end{array} \right\} \\ \left\{ \begin{array}{l} \text{COMPUTATIONAL} \\ \text{COMP} \end{array} \right\} \end{array} \right\}$$

وهذا الشكل لا يحدد كل الاختيارات الممكن استخدامها مع عبارة USAGE إنما يشمل الأشكال الأساسية والتي تعتبر أكثر استخداماً. ويجب على المبرمج حين استخدام هذه العبارة الرجوع الى مصادر الشركة الصانعة للحاسب لمعرفة الأشكال المتوفرة مع عبارة USAGE. وفي الفقرات التالية شرح عام للاختيارات الأساسية والذي يعتبر شائعاً على أغلب أنواع الحاسبات.

١-٥-٥ صيغة DISPLAY

وعبارة USAGE IS DISPLAY تعني أن الحقل الذي عُرف من قبل المبرمج في قسم البيانات سوف يمثل القيم وفق الرمز المعروف باسم EBCDIC أي أن كل بايت BYTE سوف تحتوي على رمز (حرف أو رقم) واحد. وأحياناً يطلق على طريقة التخزين هذه اسم External or Zoned Decimal.

ويمكن أن تكون عبارة DISPLAY صريحة أو ضمنية أي أن عدم ذكرها مع كلمة USAGE يعني أن الحقل المعروف في قسم البيانات مخزن على أساس رمز واحد في كل بايت.

(٥) يقصد بها الكلمات التالية :

Extended Binary Coded Decimal Interchange Code

وتجدر الملاحظة هنا الى أن طريقة التخزين هذه أساسية عند قراءة البيانات الى البرنامج من بطاقات مثقبة أو عند اخراج البيانات الى الطابعة أو الشاشة الضوئية لأنه في كلتا الحالتين يحتاج كل رمز الى مكان خاص به على وسيلة الادخال (البطاقة) أو وسيلة الاخراج (الصفحة أو الشاشة).

مثال :

Ø5 LIST - FLD PIC X (6) USAGE IS DISPLAY.

أو

Ø5 E - NUMBER PIC 9 (5) USAGE DISPLAY VALUE 1234.

وهو يعادل :

Ø5 E - NUMBER PIC 9 (5) VALUE 12345.

ففي التوصيفات أعلاه كل رقم يأخذ من الذاكرة داخل الحاسب (بايتا) واحدة.

٥ - ٥ - ٢ صيغة COMP - 3

من أجل التوفير في السعة التخزينية للأرقام عوضاً عن تخزين كل رقم في بايت واحدة كما في الشكل DISPLAY فإنه يمكن استخدام شكل أو صيغة COMP - 3 لتوصيف الحقل الرقمي على أساس أن كل بايت (أو وحدة تخزينية) يمكن أن تتسع لرقمين. وهذه الطريقة من التخزين تعرف باسم Packing أو أحياناً Packed decimal. فمثلاً الحقل الموصف في الفقرة السابقة E - NUMBER يحتاج الى ٥ مخازن (بايت) لحفظه في الحاسب بينما عند استخدام شكل COMP - 3 ينخفض العدد الى ثلاثة فقط وتظهر أهمية ذلك عند وجود أماكن تخزين داخلية في البرنامج أو على القرص المغنط رقمية كبيرة.

(هـ) من حيث المبدأ فإن الأرقام من صفر الى ٩ يمكن تمثيلها بالنظام الثنائي Binary في أربعة مواضع (بت bit) فقط. وما أن البايت تتألف عادة من ثمانية bit فإنه يمكن قسمها الى منطقتين Zones وكل منطقة تخصص لرقم.

مثال :

05 F - NUMBER PIC 99 USAGE COMP - 3 VALUE 12.

05 QNTY PIC 999 COMP - 3 VALUE 329.

فالحقلان E - NUMBER و QNTY يمكن تمثيلهما على الشكل التالي :

| Byte | | Byte | | Byte | |
|------------|---|------|---|------|---|
| 1 | 2 | 0 | 3 | 2 | 9 |
| F - NUMBER | | QNTY | | | |

وفي الشكل 3 - COMP يجب التصريح عنه إذا أراد المبرمج استخدامه وعدم ذكر هذا الشكل يعني أن الشكل DISPLAY هو المطلوب .

واستخدام 3 - COMP يعتبر مناسباً لاجراء العمليات الحسابية على الحقول الرقمية إلا أنه يجب على المبرمج أن يكون حريصاً على توصيف الحقول المعالجة من نفس الشكل أي أن لا يقوم بعملية حسابية على حقلين احدهما موصوف بـ 3 - COMP والآخر بـ DISPLAY . كما يجب الانتباه الى أن نتائج المعالجة الحسابية للشكل 3 - COMP سوف تكون أيضاً على نفس طريقة التخزين .

وتجدر الإشارة هنا الى أن الشكل 3 - COMP لا يصلح لتوصيف الحقول التي سوف تستخدم لظهار النتائج على الشاشة أو الطابعة أو لإدخال البيانات من البطاقات المثقبة . ويجب على المبرمج استعمال حقول وسيطة موصوفة بشكل DISPLAY لهذا الغرض .

وعندئذ تقوم تعليمة MOVE آلياً بتوزيع الحقل المخزن على اساس 3 - COMP الى شكل DISPLAY أو العكس تضغط حجم الحقل الموصوف في DISPLAY الى شكل 3 - COMP . أن أن تعليمة MOVE تقوم بعملية Pack و Unpack للبيانات المخزنة .

مثال :

77 MARK PIC 9 (5) COMP - 3.

77 E - MARK PIC 9 (5).

MOVE MARK TO E - MARK

DISPLAY E - MARK UPON PRINTER.

٥ - ٥ - ٣ صيغة COMP

أما عبارة USAGE IS COMP فإنها تعني أن وصف الحقل الرقمي ممثلاً على طريقة النظام الثنائي Binary. ومن مميزات استخدام هذا الشكل في تمثيل الحقل الرقمي في الحاسب امكانية التحكم في تعديل أي «بت» bit في «البايت» Byte الواحدة، وكذلك السرعة في إنجاز العمليات الحسابية على الحقول الرقمية إضافة الى سعة أكثر في التخزين (المكان المخصص للحقل الرقمي).

ويجب ملاحظة أن عملية تحويل (أو نقل) الحقل الموصوف بـ COMP الى شكل DISPLAY غير ضرورية وإنما يمكن إعادة توصيفه بوساطة عبارة REDEFINES واستخدامه في شكله الآخر بعبارة DISPLAY في عمليات الإدخال والإخراج.

مثال :

05 BNRY - F PIC 99 USAGE COMP VALUE 200.

77 BNRY - F2 PIC 999 USAGE COMP VALUE 32000.

ففي الحقل الأ ول BNRY - F فإن القيمة 200 المؤلفة من ثلاثة أرقام يمكن وضعها في مخزين (٢ بايت) لأنه حسب النظام الثنائي يحتويان على ١٦ بت (أو موضع ثنائي).

٦-٥ عبارة إعادة التسمية RENAMEs

إن عبارة إعادة التسمية RENAMEs تسمح بإعادة تجميع عدة حقول جزئية باسم مختلف عن الاسم الأول الذي عرفت به مسبقاً. وتكتب هذه العبارة في قسم البيانات. وتأخذ هذه العبارة الشكل العام التالي :

66 identifier - 1, RENAMEs identifier - 2

$$\left[\begin{array}{c} \text{THROUGH} \\ \text{THRU} \end{array} \right] \text{ identifier - 3}$$

وهذه العبارة تلي مباشرة الحقول المراد إعادة تسميتها. كما يجب أن تكون الحقول الأصلية والمسماة ثانية في نفس السجل.

وحسب قواعد لغة كوبيولر لا يمكن توصيف نفس عبارة إعادة التسمية برقم مستوى 66 ولا يمكن إعادة تسمية المستوى رقم 77 أو 88 أو 101. كما يجب أيضاً أن لا يكون identifier - 2 و identifier - 3 موصوفين بعبارة OCCURS، أو أي حقل منهما موصوف بهذه العبارة.

أما كلمة THRU وكلمة THROUGH فهما متعادلان وتعطيان نفس المعنى. وعند استخدام كلمة THRU يجب أن يكون identifier - 2 يسبق منطقياً الحقل identifier - 3.

مثال :

01 DATA - RECORD.

05 D - FLD - 1 PIC 99.

05 D - FLD - 2 PIC X (4).

(٥) Model 990 Computer COBOL Reference Manual Texas Instruments, 1980 PP. 5 - 31, 5 - 32

ϕ5 D - FLD - 3.
 ϕ7 DFLD - 3A PIC X.
 ϕ7 DFLD - 3B PIC X (1ϕ).
 ϕ5 D - FLD - 4 PIC X.
 66 D - FLD - X RENAMES D - FLD - 1 THRU DFLD - 3A.
 66 D - FLD - Y RENAMES D - FLD - 2 THRU DFLD - 3B.

 ϕ1 W - AREA.
 ϕ2 W - A PIC X.
 ϕ2 W - B PIC 9 (6).
 66 CH - CODE RENAMES W - A.

٥ - ٧ تعليمية النسخ COPY

تسمح تعليمية النسخ COPY للمبرمج أن يضع في برنامجيه أية أجزاء أو أسطر من برامج أخرى في كل من قسم المحيطات ENVIRONMENT أو قسم البيانات DATA أو في قسم الإجراءات PROCEDURE. أو بمعنى آخر تسمح تعليمية COPY بنقل بعض محتويات المكتبة Library (التي تحتوي عادة على برامج أو روتينات) المخزنة على القرص الممغنط.

وأماكن استخدام هذه التعليمية في البرنامج تظهر على النحو التالي :

1) ENVIRONMENT DIVISION

in (CONFIGURATION SECTION) :

SOURCE - COMPUTER COPY (Library - name)

OBJECT - COMPUTER COPY (Library - name)

SPECIAL - NAMES COPY (Library - name)

in (INPUT - OUTPUT SECTION) :

FILE - CONTROL COPY (Library - name)

I - O - CONTROL COPY (Library - name)

2) DATA DIVISION

in (FILE SECTION) :

FD file - name COPY (Library - name)

in a file Description :

01 data - name COPY (Library - name)

(identifier)

PROCEDURE DIVISION

paragraph - name COPY (Library - name)

أمثلة :

01 IN - REC COPY PAYROLL.

FD FILE - 1 COPY INFILE.

PA - 1. COPY LASTRTN.

والأسماء PAYROLL و INFILE و LASTRTN هي عناوين للمكتبة (اسم الملفات) التي تحتوي البيانات المراد نسخها أو نقل صورة عنها الى جانب IN - REC أو FILE - 1 أو PA - 1.

01 PAYROLL COPY PYLIB.

: حيث PYLIB

01 PYLIB.

05 F1 PIC X.

| | | |
|----|----|-------------|
| φ5 | F2 | PIC 99. |
| φ5 | F3 | PIC X (1φ). |

٥ - ٨ تجزئة قسم الاجراءات SEGMENTATION

إن تقنية تجزئة قسم الاجراءات في لغة كويول تسمح بتقسيم البرنامج (Physically) الى أجزاء وكل جزء يدعى Segment وهذا الجزء يشمل مجموعة من التعليمات المترابطة التي تقوم بإنجاز غرض معين أو وظيفة خاصة في البرنامج . Particular function

ومن أهم فوائد التجزئة هذه إعطاء الامكانية للمبرمج لتصميم البرنامج على أجزاء يستطيع التحكم بموجها في المساحة المتوفرة في الذاكرة لتنفيذ البرامج الطويلة وإبقاء الأجزاء غير اللازمة أثناء عملية تنفيذ البرنامج مخزنة على القرص المغنط (Overlays) . وبالمطبع نتيجة رخص قيمة الذاكرات في الوقت الحالي وامكانية استخدام تقنية الذاكرة المنطقية Virtual Storage فقد تقلصت أهمية هذه التجزئة وقل اعتماد المبرمج عليها بالنسبة للسنوات السابقة.

الأجزاء وترقيمها Segments

عند استخدام التجزئة في قسم الاجراءات من قبل المبرمج يمكن وضع مجموعات التعليمات في أجزاء وكل جزء يدعى Segment وكل جزء يتألف من فصول Sections وكل فصل يتألف من فقرات Paragraphs . وللدلالة على تسلسل الأجزاء فإنها تعطى أرقاماً تبدأ من φ الى الرقم ٩٩ .

(٥) TI990 يستخدم الترقيم من φ الى 127 . ص . 3 - 6

وتأخذ اسم فقرة التقسيم أو التجزئة الشكل العام التالي :

Section - name SECTION [Segment - number].

حيث أن رقم الجزء Segment - number يمكن أن يأخذ الأرقام من 1 إلى 99 . ويمكن استخدام خانة واحدة للأرقام التي تكون أقل من 10 . وإذا لم يذكر رقم الجزء بجانب كلمة الفصل SECTION فإنه يفترض أن يكون 1 .

وهذه الأرقام تأخذ دلالة معينة عند استخدامها فهي تقسم الأجزاء Segment الى نوعين :

أ - الأجزاء الثابتة Fixed or Permanent

وهي تلك الأجزاء التي تأخذ الأرقام من 1 إلى 49 وتسمى بالأجزاء الثابتة لأنها تبقى موجودة في الذاكرة طيلة مدة تنفيذ البرنامج . ولا يمكن وضع أي منها على أساس Overlay . كما يجب أن تكون متتابعة في قسم الاجراءات وليس من الضروري تسلسل أرقامها .

ب - الأجزاء المستقلة (غير الثابتة) Independent

وهي الأجزاء التي تأخذ الأرقام من 50 إلى 99 سميت بالأجزاء المستقلة لأنها تعتبر Overlays (أي متروكة جانباً على القرص الممغنط) أثناء تنفيذ البرنامج وتدخل الى الذاكرة الرئيسية عند الحاجة لها بواسطة تعليمة الاستدعاء الخاصة بها مثل PERFORM أو CALL .

وعند استخدام المبرمج لهذه التسهيلات في التجزئة عليه أن يُراعي عدم تشابه الأجزاء الثابتة والأجزاء المستقلة بتعليمة GO TO أو PERFORM وأن يكون تصميمه

للجزء المستقل على أنه جزء متكامل من الفقرات تستدعى بشكل كامل بتعليمة واحدة من قبل الجزء الثابت . كما يجب عليه أن يلحظ الاستخدام المنطقي للتعليمات التي تقوم بنقل التسلسل في البرنامج وبين الأجزاء مثل :

GO TO

PERFORM

ALTER (GO TO تعليمة لتغيير مسار)

كما ينصح قبل استخدام هذه الخاصية في التجزئة الرجوع الى مراجع الحاسب المتوفرة عن لغة كوبرول المزودة من قبل الشركة الصانعة بسبب بعض الاختلافات للاستفادة منها .

وأخيراً، وعند استخدام تجزئة قسم الاجراءات الى فصول SECTIONS وفي حالة حاجة المبرمج الاشارة الى فقرة تحمل نفس الاسم موجودة في فصلين يمكن استخدام هذا الشكل العام للدلالة عليها :

$$\text{Paragraph - name} \left[\begin{array}{c} \underline{\text{OF}} \\ \underline{\text{IN}} \end{array} \right] \text{Section - name}$$

وإذا كانت الفقرات لا تحمل نفس الاسماء في قسم الاجراءات فلا حاجة لهذه العبارة عند الرجوع الى الفقرة المطلوبة .

٥ - ٩ البرامج الفرعية والبرامج الرئيسية

تستخدم تعليمة PERFORM لتقسيم البرنامج الواحد الى أجزاء أو فقرات رئيسية وأخرى فرعية لأنها تمكن المبرمج من استدعاء أي جزء من فقرات الاجراءات بما يسهل عملية تحديد الجزء الرئيسي في البرنامج والأجزاء (الروتينات) الفرعية التي يمكن

الاستفادة منها في أكثر من موضع في البرنامج الواحد. وهذا التكوين في البرنامج يطلق عليه اسم Structured Program أي البرنامج المبنى على هياكل (هيكل أساسي وأجزاء فرعية).

أما إذا احتاج المبرمج الاستفادة من برنامج كامل آخر في البرنامج الذي يكتبه، فإن ذلك يتطلب استدعاء البرنامج الآخر إلى برنامجه الأصلي (الرئيسي) وهذا يتوفر في لغة كوبرول باستخدام تعليمة CALL (الاستدعاء) والتي تقوم بنقل تسلسل تنفيذ البرنامج الأصلي إلى البرنامج الفرعي المُستدعى CALLED ثم متابعة تنفيذ البرنامج الرئيسي بعد تعليمة CALL مباشرة.

وتأخذ تعليمة CALL الشكل العام التالي :

$$\text{CALL} \left\{ \begin{array}{l} \text{identifier - 1} \\ \text{literal - 1} \end{array} \right\} \left[\text{USING data - name - 1, data - name - 2} \dots \right] \left[\text{ON OVERFLOW imperative - Statement} \right]$$

حيث أن literal - 1 يمثل اسم البرنامج الفرعي المبين في جملة PROGRAM - ID أما استخدام مكان الاسم Identifier - 1 فإن ذلك يعين إعطاء اسم متحرك Dynamic أثناء التنفيذ.

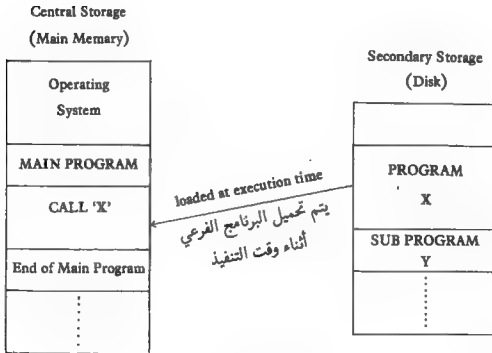
أما عبارة USING الاختيارية فهي تحدد المناطق المشتركة بين البرنامج الرئيسي والبرنامج الفرعي المستدعى وهي أسماء الحقول أو السجلات التي تغطي أو تأخذ البيانات من وإلى البرنامج الرئيسي والفرعي.

وعند وجود هذه العبارة مع تعليمة CALL يتطلب توصيف فصل خاص للحقول المشتركة مع البرنامج الرئيسي في البرنامج الفرعي و يدعى LINKAGE SECTION ●●

و يأتي في قسم توصيف البيانات (في البرنامج الفرعي) بعد فصل توصيف منطقة العمل الداخلي WORKING-STORAGE .

كما يجب ذكر اسماء الحقول المشتركة مع عبارة USING الى جانب جملة PROCEDURE DIVISION في البرنامج الفرعي أيضاً . ولا يشترط أن تكون هذه الأسماء واحدة في التسمية بل يشترط أن تكون بنفس الترتيب ونفس الحجم والنوع بين البرنامج الرئيسي والفرعي .

أما عبارة ON OVERFLOW الاختيارية أيضاً فتعني الإشارة الى نقل تسلسل تنفيذ البرنامج الرئيسي الى الفقرة الواردة فيها عند حصول «فيضان» أو زيادة في حجم البرنامج الفرعي المستدعى Called عن حجم الذاكرة المخصص للتشغيل .



الشكل ١-٥

و يتم تحميل Loading البرنامج الفرعي أثناء وقت تنفيذ البرنامج الرئيسي من الذاكرة الثانوية أو الأقراص المغنطة إلى الذاكرة الرئيسية أو الذاكرة المركزية. فإذا كان المكان المخصص في الذاكرة الرئيسية غير كاف فإن تنفيذ البرنامج الرئيسي ينتقل إلى التسلسل المبين بعد جملة ON OVERFLOW والشكل (٥ - ١) السابق يوضح عملية استدعاء البرنامج الفرعي أثناء تنفيذ البرنامج الرئيسي .

ولأجل عودة تنفيذ البرنامج الرئيسي في تسلسله المنطقي الأصلي تستخدم تعليمة EXIT PROGRAM في النهاية المنطقية للبرنامج الفرعي عوضاً من تعليمة STOPRUN . وهذه التعليمة تعادل كلمة RETURN في لغات البرجة الأخرى .

وبعد تنفيذ البرنامج الفرعي بعد استدعائه من قبل البرنامج الرئيسي فقد إشتملت لغة كويول بعد تطويرها ١٩٧٤ على تعليمة تمكن المبرمج من تخليص الذاكرة الرئيسية من البرنامج الفرعي المنفذ لإعطاء (لاتاحة) المجال لبرنامج فرعي آخر من احتلال مكانه المخصص في الذاكرة الرئيسية والتي غالباً ما تكون محدودة الحجم بالنسبة للذاكرة الثانوية . وهذه التعليمة تدعى تعليمة الإلغاء CANCEL وتأخذ الشكل العام التالي :

$$\underline{\text{CANCEL}} \left\{ \begin{array}{c} \text{identifier - 1} \\ \text{literal - 1} \end{array} \right\} \left[\left[\begin{array}{c} \text{identifier - 2} \\ \text{literal - 2} \end{array} \right] \right] \dots$$

ويقصد بالتغير identifier أو الثابت literal اسم البرنامج الفرعي المراد الغاؤه من الذاكرة الرئيسية بعد تنفيذه .

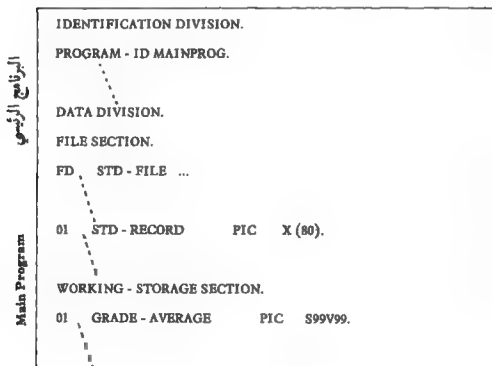
(٥) Secondary Storage or Disk

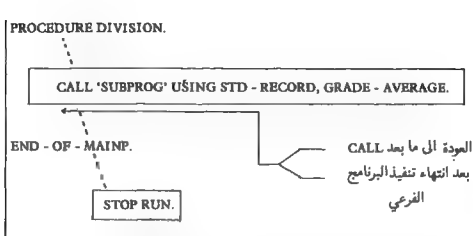
(٥٥) Main Memory or Central Storage.

وتأتي تعليمة الإلغاء بعد تعليمة الاستدعاء في البرنامج الرئيسي ، وقد لا تكون ضرورية في بعض أنواع الحاسبات حيث يقوم نظام التشغيل بعملية الإلغاء هذه بدلا من المبرمج . وينصح الرجوع الى مراجع الشركة الصانعة عند استخدام طريقة البرمجة المجزئة Modular Programming .

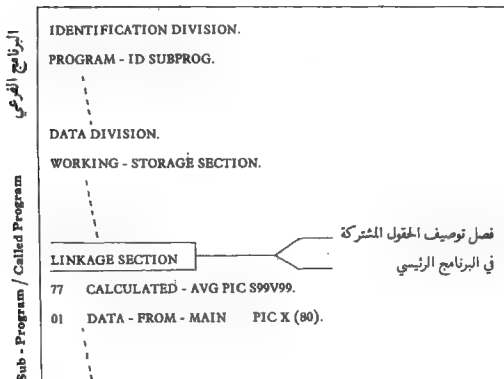
و يوجد في الصفحتين التاليتين مثال عن كيفية استخدام تعليمة الاستدعاء في البرنامج الرئيسي لبرنامج فرعي آخر يدعى SUBPROG .

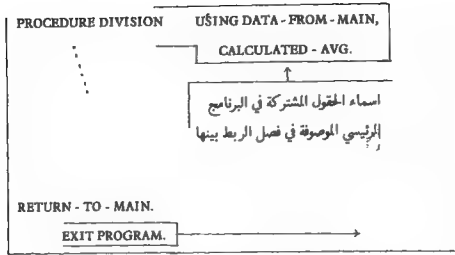
و يلاحظ من المثال أن ترتيب أسماء الحقول المشتركة بعد عبارة USING واحد في البرنامج وإن اختلفت الأسماء . كذلك فإن توصيفهما واحد وكذلك الحجم واحد . كما يلاحظ أيضاً أن البرنامج الرئيسي ينتهي منطقياً بتعليمة STOP RUN والبرنامج الفرعي بتعليمة EXIT PROGRAM .





استدعاء البرنامج الفرعي المسمى SUBPROG مع استخدام الحقول المشتركة والمسماة
في البرنامج الرئيسي STD - RECORD, GRADE - AVERAGE .





وفي ختام هذا الفصل لا بد من الإشارة الى أن عملية وجود برنامج رئيسي وبرامج فرعية أخرى مستدعاة من قبله تتطلب عملية ترجمة كل برنامج على حدة وبشكل مستقل ثم تحتاج الى عملية ربط منطقي بينهما لأجل تعريف الحاسب بأن البرنامج الرئيسي أثناء تنفيذه يحتاج الى برامج (أو برنامج) فرعية أخرى لذلك لا بد من إجراء عملية بعد الترجمة **COMPILATION** وتسمى بعملية الربط المنطقي **Linkedit** وهذه العملية تتم بعد خلو البرامج الفرعية والرئيسية من الأخطاء وتصبح جاهزة للاختبار أو التشغيل.

مثال عملي

في الصفحات التالية نموذج عن برنامج رئيسي وبرنامجان فرعيان تم اعدادهما وتنفيذهما على الحاسب **IBM** لأجل إيضاح كيفية ربط البرامج مع بعضها حيث أن أحد هذين البرنامجين الفرعيين يقوم بعملية الضرب (التربيع) والآخر يقوم بعملية الجمع على نفس القيم و ثم تظهر النتيجة بوساطة تعليمات البرنامج الرئيسي .

(هـ) هناك بعض أنواع من الحاسبات لا تحتاج الى عملية الربط هذه وإنما تنفذ آلياً من قبل نظام التشغيل مثل **IMOS** **NCR** لذلك ينصح بالرجوع الى مراجع الشركة الصانعة .

```

IDENTIFICATION DIVISION.
PROGRAM-ID.      MAINP.
AUTHOR. AJND-KHAYAT.
**          THIS PROGRAM IS TO ILLUSTRATE **
**          COBOL CALL STATEMENT.      **
**          IT IS MAIN PROGRAM          **
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-3033.
OBJECT-COMPUTER. IBM-3033.
DATA DIVISION.
WORKING-STORAGE SECTION.
77  FUNC          PIC X.
77  WAIT1         PIC X.
01  AREA-S.
    05 MA1        PIC 99999.
    05 MB1        PIC 9(11).
*
PROCEDURE DIVISION.
ST.
    DISPLAY 'ENTER 5 DIGIT-NUMBER '.
    ACCEPT MA1
    IF MA1 NOT NUMERIC
        MOVE ZEROES TO MA1.
    DISPLAY
    'SELECT (M) FOR SQUARE '
    'OR (A) FOR ADD '
    'OR (9) FOR END '.
    ACCEPT FUNC.
    IF FUNC = 'M'
        CALL 'SPROG' USING AREA-S
        DISPLAY 'THE SQUARE VALUE IS '
        DISPLAY MB1
    ELSE
    IF FUNC = 'A'
        CALL 'SPROG2' USING AREA-S
        DISPLAY 'THE TOTAL VALUE IS '
        DISPLAY MB1
    ELSE
    IF FUNC = '9' STOP RUN.
    GO TO ST.

```

البرنامج الرئيسي
Main Program

```

IDENTIFICATION DIVISION.
PROGRAM-ID.      SPROG.
AUTHOR.  AJND-KHAYAT.
**      THIS PROGRAM IS TO ILLUSTRATE  **
**      COBOL CALL STATEMENT,         **
**      IT IS SUBPROGRAM FOR SQUARE  **
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.  IBM-3033.
OBJECT-COMPUTER.  IBM-3033.
DATA DIVISION.
WORKING-STORAGE SECTION.
01  AREA-S.
    05  A1          PIC 99999.
    05  B1          PIC 9(11).
LINKAGE SECTION.
01  SAREA.
    05  SA1         PIC 99999.
    05  SB1         PIC 9(11).
*
PROCEDURE DIVISION USING SAREA.
ST.
    MULTIPLY SA1 BY SA1 GIVING SB1.
    DISPLAY 'SUBPROGRAM SPROG EXECUTED'.
EXTP.
    EXIT PROGRAM.

```

البرنامج الفرعي
Subprograms

```

IDENTIFICATION DIVISION.
PROGRAM-ID.      SPROG2.
AUTHOR.  AJND-KHAYAT.
**          THIS PROGRAM IS TO ILLUSTRATE  **
**          COBOL CALL STATEMENT.         **
**          IT IS SUBPROGRAM FOR ADITION  **
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.  IBM-3033.
OBJECT-COMPUTER.  IBM-3033.
DATA DIVISION.
WORKING-STORAGE SECTION.
01  AREA-S.
    05  A1          PIC 99999.
    05  B1          PIC 9(11).
LINKAGE SECTION.
01  SAREA.
    05  SA1         PIC 99999.
    05  SB1         PIC 9(11).
*
PROCEDURE DIVISION USING SAREA.
ST.
    ADD      SA1,      SA1  GIVING SB1.
    DISPLAY  'SUBPROGRAM SPROG2 EXECUTED'.
EXTP.
    EXIT PROGRAM.
*

```

٥- ١٠ اسئلة وتمارين

سؤال ١- بين الخطأ والصحيح من العبارات التالية بالإشارة (س) أو (x) على التوالي :

- لا يمكن استخدام عبارة REDEFINES مع أي رقم مستوى Level Number
- يمكن أن تأتي عبارة REDEFINES مع رقم المستوى 77
- ترد عبارة RENAMES مع أي رقم مستوى Level Number .
- تحتوي الفقرة Paragraph في قسم الاجراءات على فصل واحد أو أكثر و يسمى SECTION .
- تنفيذ عملية تجزئة قسم الاجراءات Segmentation في الاستغلال الأمثل للذاكرة المحدودة الحجم في تشغيل البرامج الكبيرة .
- يمكن أن ترد تعليمة CALL في البرنامج الفرعي Subprogram لاستدعاء برنامج فرعي آخر .
- يذكر فصل الربط LINKAGE SECTION في البرنامج الرئيسي Main .
- يمكن استدعاء أكثر من برنامج فرعي بواسطة عدة تعليمات CALL تذكر في البرنامج الرئيسي .
- يسمح بذكر عبارة COPY في قسم البيانات فقط .

سؤال ٢- يبين الجدول التالي توصيف وقيمة الحقل المراد نقله بواسطة تعليمة MOVE الى الحقل المسمى Receiving field . والمطلوب توضيح شكل القيمة الناتجة بعد

عملية النقل :

| Sending Field | | Receiving Field | |
|---------------|--------|-----------------|---------------|
| PICTURE | VALUE | PICTURE | EDITED RESULT |
| S9(4)V99 | -456φφ | \$\$\$\$.99CR | |
| S9(4)V99 | 456φφ | \$\$\$\$.99DB | |
| S9(4) | 4567 | \$.\$\$\$φφ | |
| S9(6) | 121082 | 99B99B99 | |
| S9(6) | 121082 | 99/99/99 | |
| S9(6) | 123456 | _____ | |
| 9(6)V99 | 123450 | \$ZZZ,ZZZ.99 | |
| 9(6)V99 | 123450 | \$ooo,ooo99. | |
| V99 | 12 | VPFP99 | |

سؤال ٣ - في الشكل التالي المأخوذ من برنامج كوبول يوجد توصيف الحقول تم إعادة توصيف لبعضها والمطلوب معرفة :

١ - حجم السجل أو الحقل AREA - A بالبايت

٢ - ما هي الحقول التي تأخذ نفس المساحة التي يمثلها الحقل P - FLD.

φ1 A - AREA.

φ2 B - AREA.

φ5 C - FLD PIC xx.

φ5 D - FLD PIC xxx.

φ2 E - AREA REDEFINES B - AREA PIC 9 (5).

Ø2 F - AREA.

Ø5 G - FLD PIC 9 (5).

Ø5 H - FLD.

Ø7 I - FLD PIC xx.

Ø7 J - FLD PIC 99.

Ø5 K - FLD REDEFINES H - FLD.

Ø7 L - FLD PIC x.

Ø7 M - FLD PIC 9 (6) COMP - 3.

Ø2 N - AREA REDEFINES F - AREA.

Ø5 Q - FLD PIC X (4).

Ø5 P - FLD PIC X (5).

تمرين ١

أكتب برنامجاً بلغة كوبيول يقرأ ملف المشتركين SUBFILE ويعمل تحققاً للحقول Validation وفق القواعد التالية :

- رقم المشترك D - No يجب أن يكون رقمياً واكبر من الصفر.
- اسم المشترك D - NAME يجب ألا يكون رقمياً، أو فارغاً.
- عنوان المشترك D - ADRESS يجب ألا يكون فراغاً.
- تاريخ الاشتراك D - DATE :
- السنة D - YEAR يجب أن تكون بين ١٤٠٢ و ١٤٠٥
- الشهر D - MONTH يجب أن يكون بين ١ - ١٢
- اليوم D - DAY يجب أن يكون بين ١ - ٣٠

- رمز الاشتراك D - CODE يجب أن يكون بين ١ - ١٠
 - منطقة الاشتراك D - AREA يجب أن تكون محصورة بين ١ - ١٥
 - مقدار القراءة D - LREAD يجب أن تكون رقمية
 - رقم العدد D - CNTR رقم العدد يجب أن يكون اكبر من الصفر
- عند وجود خطأ في أي حقل من الحقول السابقة حرك محتوى السجل الى الطابعة وأطبعه وأطبع نجمة * تحت الحقل الخطأ، وكذلك انقل محتوى السجل نفسه الى ملف اخراج آخر اسمه INVFILE موجود على قرص ممغنط. وأعد معالجة سجل جديد.
- أما اذا كانت كل الحقول سليمة انقل محتوى السجل الى ملف اخراج اسمه VALFILE على القرص الممغنط ودون طباعته، وأعد معالجة سجل جديد.

ملاحظات الطباعة :

- ١- لا تكتب في الصفحة أكثر من ٢٥ سجلاً
- ٢- اترك سطرًا فارغاً بين كل سطري طباعة
- ٣- اكتب العناوين والتاريخ ورقم الصفحة في رأس كل صفحة

توصيف سجل ملف INVFILE :

| <u>FIELD NAME</u> | <u>PICTURE</u> |
|-------------------|----------------|
| INV - NO | 9 (5) |
| INV - NAME | X (24) |
| INV - ADRESS | X(24) |
| INV - DATE | 9 (8). |
| INV - CODE | . 99 |
| INV - AREA | 99 |
| INV - CNTR | 9 (7) |

| | |
|-------------|-------|
| INV - LREAD | 9 (7) |
| INV - FLR | X (5) |

: VALFILE توصیف سجل ملف

| <u>FIELD NAME</u> | <u>PICTURE</u> |
|-------------------|----------------|
| VAL - NO | 9 (5) |
| VAL - NAME | X (24) |
| VAL - ADRESS | X (24) |
| VAL - DATE | 9 (8) |
| VAL - CODE | 99 |
| VAL - AREA | 99 |
| VAL - CNTR | 9 (7) |
| VAL - LREAD | 9 (7) |
| FLR | X (5) |

: SUBFILE توصیف سجل ملف

| <u>FIELD NAME</u> | <u>PICTURE</u> |
|-------------------|----------------|
| D - NO | 9 (5) |
| D - NAME | X (24) |
| D - ADRESS | X (24) |
| D - DATE | |
| D - YEAR | 9 (4) |
| D - MONTH | 99 |
| D - DAY | 99 |
| D - CODE | 99 |

| | | |
|-----------|-------|-----------------------------|
| D - AREA | 99 | أنظر صفحة ٦٠٣ لتوصيف شكل |
| D - CNTR | 9 (7) | المخرجات وصفحة ٦٠٠ للنتائج. |
| D - LREAD | 9 (7) | |
| D - FLR | X (5) | |

تمرين ٢

ملف رواتب يحتوي على سجلات الموظفين العائدين من الإجازة وموجودة على قرص ممغنط ويحوي المعلومات التالية :

| | FIELD NAME | PICTURE |
|----------------|-------------|-----------|
| رقم الموظف | EMP - NO | 9 (5) |
| الاسم | EMP - NAME | X (34) |
| راتبه الأساسي | EMP - SAL | 9 (5) V99 |
| بدل نقل شهري | EMP - TRANS | 9 (3) V99 |
| بدل تدريب شهري | EMP - ALW | 9(5)V99 |
| تاريخ العودة | EMP - DATE | |
| من الإجازة | | |
| يوم | EMP - DAY | 99 |
| شهر | EMP - MONTH | 99 |
| سنة | EMP - YEAR | 9(4) |
| غير مستعمل | FILLER | X (18) |

المطلوب كتابة برنامج يجري العمليات التالية :

- ١ - حساب صافي الراتب بعد احتساب الراتب الأساسي بدل النقل بدل التدريب وذلك عن المدة المتبقية من الشهر (يعتبر الشهر ٣٠ يوماً).

- وكذلك حساب التقاعد بواقع ٩% عن المبلغ المستحق من الراتب الأساسي .
- ٢ - طباعة جدول الرواتب .
 - ٣ - استعمل برنامجاً فرعياً SUBPROGRAM لإجراء جميع العمليات الحسابية .
 - ٤ - استخدم تعليمة CALL في البرنامج الأساسي من أجل استدعاء البرنامج الفرعي .
- لا تطبع في الصفحة الواحدة أكثر من ٣٠ اسم، أترك فراغاً بين كل سطري طباعة .
- أنظر صفحة ٦٠٤ لتوصيف شكل المخرجات .

الفصل السادس

الجداول في لغة كوبرول

TABLES

٦ - ١ لمحة عامة

يمكن تعريف الجدول Table • بأنه عبارة عن رموز • مرتبة بطريقة يتكرر فيها نفس الحقل عدة مرات متتالية .

فمثلاً هناك عدة بنود في فاتورة يمكن وصفها على شكل جدول حيث أن البند يتكرر في الفاتورة أكثر من مرة وقد يحتوي هذا البند على رموز السلعة أو وصفها مع السعر والقيمة . كذلك يمكن وضع رموز أسماء المناطق أو المدن مع أسمائها على شكل جدول
مثل :

- ① RIYADH
- ② JEDDAH
- ③ DAMMAM

حيث أن البند ••• (أو الصف) الواحد من الجدول يقصد به رمز المدينة مع اسمها وذلك لتسهيل عملية تداول ومعالجة هذه البيانات الموجودة في الجدول ببرنامج الكوبرول . وسوف يظهر في الصفحات التالية بعد شرح لوصف وكيفية استخدام

(•) Programmer COBOL Guide

(••) يقصد بها Characters أي أحرف وأرقام .

(•••) table entry أو table item

الجداول في الكوبول أهمية الجدولة في تسهيل عمل المبرمج واختصار العديد من الخطوات التي قد يحتاجها بدون عملية الجدولة لهذه البيانات وخاصة تلك التي يمكن صياغتها بشكل سهل وبسيط على أشكال جداول .

وتقسم أحيانا الجداول حسب أبعادها فهناك بعد واحد (أي صفوف فقط) وهناك بعدين (صف وعمود) أو ثلاثة أبعاد (صف وعمود وارتفاع كما في أشكال الحجم).

وقبل الدخول في الأشكال الخاصة بكوبول لا بد من نوه الى أن الصف أو العمود في الجدول الخاص بكوبول يمكن أن يحتوي على حقل أو أكثر data - name . كما أن هناك طريقتين أو اسلوبين مختلفين في كوبول لمعالجة وتداول بيانات الجدول وهما :

- اسلوب :
- استخدام المؤشر Subscripting
 - استخدام الدليل Indexing

ولكل من هذين الاسلوبين مزاياه ولكن لا بد للمبرمج من التمييز بينهما حيث أنهما متشابهان أو مترادفان في لغات البرمجة الأخرى .

تعريف الجدول وعبارة التكرار OCCURS

يقصد بكلمة OCCURS ه في هذا المجال تكرار حدث معين مرات عديدة بنفس الشكل ، وهذا ما يمكن تطبيقه على عملية تكرار نفس السطر أو العمود في الجدول لأنه يأخذ نفس الشكل ، لذلك فإن لغة كوبول تستخدم عبارة OCCURS للدلالة على

(ه) تعني OCCURS يوجد أو يحدث ، يظهر (المورد طبعة ١٩٨٢)

تعريف الجدول في قسم البيانات . وتأخذ هذه العبارة عدة صيغ عامة نذكر أبسطها وهي الصيغة التالية :

(Level - number) (data - name) OCCURS (integer) TIMES.

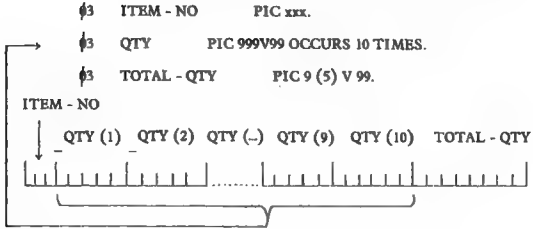
ورقم المستوى Level - number يمكن أن يكون أي رقم مستوى عدا 01 أو 77 أو 88 أو 66 أي ينحصر بين 02 و 49 ، ويمكن استخدام عبارة OCCURS لتوصيف أي جزء من السجل Record الخاص بالملف ولكن لا يمكن استخدامها لتوصيف سجل أو بند Item مستقل بذاته .

أما عدد مرات تكرار الصف أو البند في الجدول فهو يمثل في العدد الصحيح Integer ويجب أن يكون عدداً موجباً ويختلف الحد الأقصى له باختلاف نوع الحاسب المستخدم فبعضها يعمل إلى 32767 .

كما أن عبارة OCCURS تصلح لتعريف الحقل أو البند الأولي ELEMENTARY أو مجموعة من الحقول GROUP . على أن لا تحتوي عبارة OCCURS عبارة VALUE المستخدمة لتوصيف القيمة الثابتة . وتجدر الإشارة إلى أنه لا يمكن استخدام إعادة تعريف الحقل الموصوف بعبارة OCCURS بواسطة عبارة إعادة التعريف REDEFINES بل العكس جائز ويمكن . أي إعادة تعريف أي حقل باسم آخر مع عبارة OCCURS .

(*) The Programmer's ANSI COBOL, Ref. Manual Donald A. Sordillo, Prentice - Hall 1978 PP. 166

الشكل التالي يبين استخدام OCCURS في التعريف :



تعريف الجدول ذو أكثر من بُعد NESTED OCCURS

إن الجدول الموصوف بعبارة واحدة OCCURS يعتبر جدولاً ذا بُعد واحد وظهور عبارة تالية OCCURS في مستوى Level أدنى وأقل تابع للعبارة الأولى يدل على وجود جدول فرعي آخر يتبع للجدول الأول . وهذا ما يسمى بالجدول ذي البعدين (صف وعمود) أو هناك في الجدول نوعان من التقسيمات : تقسيمات إجمالية مثلاً : أقسام في مؤسسة ، ثم تقسيمات فرعية : مثلاً : شعب متعددة تتبع هذه الأقسام . فمثلاً الشكل التالي بجدول يحتوي أربعة أقسام للبرامج التدريبية في معهد الإدارة العامة وكل قسم يحتوي على عدد من البرامج يصل في المثال هذا إلى خمسة برامج فيمكن توصيف هذا الجدول كما يلي :

| القسم (4) برامج تدريبية | القسم (3) برامج خاصة | القسم (2) برامج اعدادية | القسم (1) برامج عليا |
|--|--|--|--|
| (1) (2) (3) (4) (5) (1) (2) (3) (4) (5) | (1) (2) (3) (4) (5) (1) (2) (3) (4) (5) | (1) (2) (3) (4) (5) (1) (2) (3) (4) (5) | (1) (2) (3) (4) (5) (1) (2) (3) (4) (5) |

Ø1 EXAMPLE - TAB - 2.

Ø5 DIVISIONS OCCURS 4 TIMES.

Ø1 PROGRAMS OCCURS 5 TIMES PIC 99.

أما إذا تواجد في توصيف الجدول ثلاثة عبارات لـ OCCURS وكل واحدة منها تابعة للسابقة لها في تسلسل رقم المستوى فعندئذ يكون هذا الجدول ذا ثلاثة أبعاد . والجدول التالي يعطي صورة لجدول مكون من ثلاثة أبعاد :

| FLDX | FLDY (1) | | | | FLDY (2) | | | | FLDZ |
|----------|--------------|-----|--------------|-----|--------------|-----|--------------|-----|------|
| | FLDY - A (1) | | FLDY - A (2) | | FLDY - A (3) | | FLDY - A (4) | | |
| | | | | | | | | | |
| | (1) | (2) | (1) | (2) | (1) | (2) | (1) | (2) | |
| FLDY - B | | | | | | | | | |

ويمكن توصيفه كما يلي :

Ø1 EXAMPLE - TAB - 3.

Ø5 FLDX PIC XXX.

Ø5 FLDY OCCURS 2 TIMES.

1Ø FLDY - A OCCURS 2 TIMES.

15 FLDY - B OCCURS 2 TIMES.

2Ø FLDYB1 PIC X.

2Ø FLDYB 2 PIC XX.

Ø5 FLDZ PIC XXX.

(هـ) لا يستخدم في لغة كوبرل أكثر من ثلاثة أبعاد.

و يلاحظ من الشكل السابق أن عدد تكرار حدوث البنود FLDY هو مرتان (1)
 FLDY و (2) FLDY أي أن FLDY هو جدول ذو بعد واحد أما الحقل أو البند
 FLDY - A فهو مكرر مرتين ضمن الجدول الأصلي FLDY أن أن هناك أربعة حقول
 جزئية كما يلي :

$$FLDY - A (1,1), FLDY - A (1,2) FLDY - A (2,1) FLDY - A (2,2)$$

والأرقام الموجودة بين قوسين تدل على تسلسل أو ترتيب العنصر في الجدول فالعنصر
 (أو البند) FLDY - A (1,2) يعني أن العنصر الأول من الجدول الأول أي أول عبارة
 OCCURS وهو في نفس الوقت العنصر الثاني من الجدول الفرعي أي ثاني عبارة
 OCCURS أو كما يوصف في عمليات المصفوفات بدلالة السطر والعمود على التوالي .
 وعند أخذ عبارة OCCURS الثالثة في الاعتبار يصبح لدينا البنود الجزئية التالية :

$$FLDY - B (1,1,1) , FLDY - B (1,1,2)$$

$$FLDY - B (1,2,1) , FLDY - B (1,2,2)$$

$$FLDY - B (2,1,1) , FLDY - B (2,1,2)$$

$$FLDY - B (2,2,1) , FLDY - B (2,2,2)$$

حيث أنه كل بند من هذه البنود يحتوي حقلين جزئيين وهما :

$$FLDY - B1 (I,J,K)$$

$$FLDY - B2 (I,J,K) \text{ و}$$

وللإشارة الى أي عنصر أو بند في الجدول سواء كان ذا بُعد واحد أو أكثر تستخدم
 الأقواس والرقم الخاص بترتيب هذا البند أو العنصر بينهما و يطلق على هذا الرقم اسم
 المؤشر أو الدليل Subscript أو Index وهناك اختلاف بين الاثنين في لغة كوبرول كما
 سوف يلاحظ في الصفحات التالية .

(هـ) قد يستعمل أحيانا باسم الرمز أو الدليل السفلي لأنه يذكر في رموز الرياضيات في أسفل وجانب المتغير مثل : س_١ ،
 س_٢ س_٣ ...

٦- ٤ تداول الجداول ACCESSING TABLES

تتم عملية تداول عناصر الجدول المعروف بعبارة أو عبارات OCCURS بواسطة تحديد ترتيب وموضع هذا العنصر في الجدول فإذا كان الجدول ذا بُعد واحد يحدد رقم السطر أو البند الذي يحتوي العنصر وهذا الرقم نسبي يبدأ من الرقم ١ إلى نهاية حجم الجدول. وإذا كان الجدول ذا بُعدين يحدد رقم السطر ثم رقم العمود الذي يدل على العنصر أو البند في الجدول وكذلك هذه الأرقام نسبية لأنها تبدأ من الرقم ١ إلى نهاية حجم الجدول. وكذلك إذا كان الجدول ذا ثلاثة أبعاد تحدد المؤشرات الثلاثة التي تبين موضع العنصر في الجدول، لأن جميع عناصر الجدول المكررة في عبارة OCCURS تحمل نفس الاسم ولأجل تمييزها عن بعضها يشار إليها بموضعها النسبي في الجدول Relative Position، وتوجد طريقتان في لغة كوبيول من أجل تداول عناصر الجدول وتحديد موضع هذه العناصر وهما :

- استخدام المؤشر Subscripting
- استخدام الدليل Indexing

٦- ٤- ١ استخدام المؤشر SUBSCRIPTING

بعد توصيف الجدول Table ومن أجل الإشارة إلى عنصر أو بند في هذا الجدول يستخدم حقل رقمي يدعى المؤشر Subscript عُرفاً ويوصف في قسم البيانات في فصل التوصيف الداخلي في البرنامج كأى حقل آخر مع ملاحظة أن حجمه يجب أن يتناسب مع حجم أو عدد العناصر المكونة للجدول. كما يمكن الدلالة على المؤشر مباشرة باستخدام رقم ثابت صحيح يمثل المكان النسبي للعنصر في الجدول ومن أجل تمييزه عن غيره من الثوابت أو الحقول يوضع بين قوسين بعد ذكر اسم البند المكرر في الجدول

مثل: QTY (1) , QTY (5)

للدلالة على العنصر رقم (1) والعنصر رقم (5) في الجدول الموصوف في الصفحات السابقة.

أو عند استخدام حقل وسيط لتخزين مكان العنصر مثل الاسم SUB1 يكون:

WORKING - STORAG SECTION.

77 SUB1 PIC 99.

PROCEDURE DIVISION.

MOVE 1 TO SUB1

MOVE QTY (SUB1) TO

MOVE 5 TO SUB1

MOVE QTY (SUB1) TO

وإذا كان الجدول ذا أكثر من بُعد واحد يكون الشكل التالي للدلالة على العنصر

فيه :

FLDY - A (2,1)

أو FLDY - A (S1,C1)

حيث يدل الرقم الأول على رقم السطر والرقم التالي على رقم العمود في الجدول

ويأخذ استخدام المؤشر للدلالة على عنصر الجدول الشكل العام التالي :

* { data - name - 1 } [{ OF } data - name - 2]
Condition - name [{ IN }

(Subscript - 1 [, Subscript - 2 [, Subscript - 3]])

(*) لا يسمح نظام T1990 بوجود Condition - name

حيث أن المؤشر يوضع بين قوسين و يوجد فراغ قبل وآخر بعد القوس الثاني للفصل بينه وبين أسماء الحقول المجاورة. وإذا وُجد أكثر من مؤشر (جدول ذو أكثر من بُعد) توضع المؤشرات على الترتيب حسب التسلسل المنطقي لوضع العنصر في الجدول و يفصل بينها فاصلة أو فراغ * .

كما تجب الإشارة الى أن المؤشر لا يمكن أن يكون مقبولا إذا كان هو بعد ذاته جدولا ويحتاج الى مؤشر آخر له. أي يكون Subscripted Subscript .

وفي حالة كون المؤشر المستخدم عدد غير ثابت أي اسم الحقل فإنه يمكن إجراء العمليات الحسابية عليه وكذلك تعليمات المقارنة IF.

كما أن هناك بعض أنواع من الحسابات * تسمح بأن يكون هذا المؤشر عدداً جبرياً أي يحتوي على إشارة جبرية إلا أنه يجب أن لا يقل عن قيمة 1 وأن يكون عدداً موجب الإشارة.

٦-٤-٢ استخدام الدليل INDEXING

إن استخدام الدليل أو الفهرس Index يشبه الى حد كبير استخدام المؤشر من أجل الدلالة على موضع العنصر في الجدول وتداول مع بعض الاختلافات ويمكن تلخيصها فيما يلي :

- يضاف الى عبارة OCCURS جملة ثانية هي INDEXED BY من أجل تحديد اسم الدليل الخاص بالجدول المَعرّف بـ OCCURS و يصبح الشكل العام لهذه العبارة :
(level - number) (date - name) OCCURS (integer) TIMES
[INDEXED BY index - name - 1 [index - name - 2]]

حيث أن كلمة index - name تدل على اسم يحدد من قبل المبرمج للدلالة على الدليل المستخدم مع عبارة OCCURS وهذا الدليل عبارة عن حقل يوصف آلياً من قبل نظام التشغيل (مترجم كوبول) ولا حاجة لتعريفه من قبل المبرمج في قسم البيانات ويتعامل نظام التشغيل معه بقيمة العدد الثنائي Binary Value من أجل تحديد مواضع العناصر في الجدول .

- تستخدم تعليمة خاصة تدعى SET لزيادة أو تخفيض قيمة الدليل Index - name وكذلك لاعطائه القيمة الأولية (الابتدائية) ولا يمكن استخدام التعليمات الحسابية ... , SUBTRACT, ADD أو تعليمة النقل MOVE لأجل تعديل قيمته .
 - يجب استخدام نفس أسماء الأدلة المذكورة مع عبارة OCCURS للدلالة على العنصر في الجدول ولا يجوز استخدام أسماء حقول أو متغيرات أخرى ولذلك يجب تحديد قيمة هذا الدليل بخطوة سابقة لتعليمة تداول العنصر في الجدول .
 - يمكن استخدام الجدول الموصف مع دليل Index - name بتعليمة SEARCH أما وجود المؤشر Subscript فهو لا يسمح باستعمالها .
- ويمكن إعادة توصيف الجدول المذكور في الفقرة السابقة باستخدام الدليل على الشكل التالي :

❶ EXAMPLE - TAB - X3.

❶ FLDX PIC xxx.

❶ FLDY OCCURS 2 TIMES

INDEXED BY XD1

1❶ FLDY - A OCCURS 2 TIMES

INDEXED BY XD2

15 FLDY - B OCCURS 2 TIMES

INDEXED BY XD3

2# FLDYB1 PIC X.

2# FLDYB2 PIC XX.

05 FLDZ PIC XXX.

حيث أن XD1 , XD2 , XD3 تكون أسماء لأدلة الجدول فإذا أردنا تداول أي بند في الجدول فيجب ذكر هذا الدليل للدلالة على موضع هذا البند مثل :

FLDY. (XD1)

FLDY - A (XD1 , XD2)

FLDY - B (XD1, XD2, XD3)

ولأجل إعطاء قيمة لهذه الأدلة تستخدم سابقاً لها تعليمة SET وسيأتي ذكرها في الفقرة التالية .

عبارة USAGE IS INDEX

لقد مرّ سابقاً أن عبارة USAGE IS توصف الحقل في ثلاثة أشكال أو صيغ أساسية وفي استخدام الدليل يضاف إليها الشكل الرابع INDEX و يصبح الشكل العام لهذه العبارة على النحو التالي :

$$\text{USAGE IS } \left\{ \begin{array}{l} \underline{\text{DISPLAY}} \\ \underline{\text{COMP}} \\ \underline{\text{COMP} - 3} \\ \underline{\text{INDEX}} \end{array} \right\}$$

وعبارة USAGE IS INDEX تعرّف الحقل data - name الموصوف بها على أنه حقل في صورة دليل يستخدم لأجل تخزين قيمة الدليل المذكورة بعد عبارة OCCURS بشكل مؤقت لأن كلمة INDEX بعد USAGE تعطيه (أي الحقل) نفس صورة الدليل

الأصلية أي ثنائي القيمة Binary . ولا يحدد المبرمج حجم الحقل الموصوف كدليل بل يترك ذلك الى نظام التشغيل المستخدم (أو الحاسب المستخدم) لأن حجمه يختلف باختلاف الحاسب .

مثال :

77 INDEX - STORE USAGE IS INDEX

فالحقل INDEX - STORE يأخذ نفس صيغة الدليل المعروف بعد عبارة INDEXED BY . ويعرف باسم Index data item ويمكن أن يكون موصوفاً ضمن مجموعة من الحقول Group ، أو يوصف مجموعة أدلة .

و يكون الشكل العام لاستخراج الدليل Index مع عناصر الجدول وفق الشكل التالي :

$$\begin{aligned}
 & ** \left\{ \begin{array}{l} \text{data - name - 1} \\ \text{Condition - name} \end{array} \right\} \left[\left\{ \begin{array}{l} \underline{\text{OF}} \\ \underline{\text{IN}} \end{array} \right\} \begin{array}{l} \text{data - name - 2} \\ \dots \end{array} \right] \\
 & \left(\left\{ \begin{array}{l} \text{index - name - 1} \\ \text{literal - 1} \end{array} \right\} \left[\begin{array}{l} \{ + \\ - \} \end{array} \right] \begin{array}{l} \text{literal - 2} \\ \end{array} \right] \\
 & \left[, \left\{ \begin{array}{l} \text{index - name - 2} \\ \text{literal - 3} \end{array} \right\} \left[\begin{array}{l} \{ + \\ - \} \end{array} \right] \begin{array}{l} \text{literal - 4} \\ \end{array} \right] \right] \\
 & \left[, \left\{ \begin{array}{l} \text{index - name - 3} \\ \text{literal - 5} \end{array} \right\} \left[\begin{array}{l} \{ + \\ - \} \end{array} \right] \begin{array}{l} \text{literal - 6} \\ \end{array} \right] \right] \left. \right)
 \end{aligned}$$

(*) يأخذ أربعة بايت ثنائية القيمة في نظام التشغيل NCR IMOS صفحة ٩٧ .

(**) Donald A. Sordillo P. 100 - 101

لا يوجد Condition - name في نظام TI 990

حيث يدل index - name على اسم الدليل المستخدم مع عبارة OCCURS المصحوبة
بجملة INDEXED BY.

الفهرسة المباشرة DIRECT INDEXING

و يقصد بهذه الطريقة تداول عنصر الجدول مع ذكر اسم الدليل الخاص به (أو
الأدلة الخاصة به) بين قوسين تماماً مثل استخدام المؤشر في الفقرات السابقة.

فمثلاً : FLDY (XD1)

تشير الى العنصر (أو البند) FLDY الذي موضعه في الجدول XD1.

فإذا كانت قيمة XD1 تساوي 2 فإن العنصر (2) FLDY سيكون محل التداول في
التعليمة المستخدمة وبشكل مباشر.

الفهرسة النسبية * RELATIVE INDEXING

وتعني هذه الطريقة من استخدام الدليل إضافة أو طرح قيمة ثابتة عديدة من قيمة
الدليل الموجود بين القوسين للدلالة على مكان العنصر (أو البند) في الجدول . ويجب أن
تكون عملية الإضافة أو الطرح من الدليل متفقة مع الشكل العام المذكور في الفقرة
السابقة (يأتي العدد الثابت بعد الدليل وتفصل بينهما الإشارة الحسائية + أو - وتقع بين
قراغين). فمثلاً :

إذا كان الدليل XD1 يساوي 1

FLDY (XD1 + 2)

يقصد بهذا الاسم العنصر (أو البند) FLDY رقم 3 في الجدول . أما :

FLDY (XD1 - 1)

فيقصد به العنصر أو البند رقم 2 إذا كانت قيمة الدليل XD1 تساوي 3 .
ويجب هنا الانتباه الى عدم الوصول الى القيمة 0 (صفر) أثناء الطرح .
كما أن وجود أكثر من دليل بين القوسين يجب أن يكون حسب الترتيب المنطقي
للجدول .

٦ - ٥ تعليمة تحديد قيمة الدليل *SET

تستخدم تعليمة SET لإعطاء قيمة عددية للدليل INDEX الموصوف بعبارة
INDEXED BY أو لأجل نقل قيمة هذا الدليل الى حقل آخر مسمى كدليل بعبارة
USAGE IS INDEX . وكذلك لتعديل قيمة الدليل أيضاً زيادة أو نقصان من أجل
تداول العناصر المكررة في الجدول المطلوب .

وتأخذ تعليمة SET الشكل العام التالي وعلى صيغتين :

$$1) \quad \underline{\text{SET}} \quad \left\{ \begin{array}{l} \text{index - name - 1 [index - name - 2]} \\ \text{identifier - 1 [identifier - 2]} \end{array} \right\} \dots \underline{\text{TO}} \quad \left\{ \begin{array}{l} \text{index - name - 3} \\ \text{identifier - 3} \\ \text{literal - 1} \end{array} \right\}$$

حيث أن 1 - index - name و 2 - index - name ...

هي أسماء للأدلة المذكورة بعبارة OCCURS مع جملة INDEXED BY أما
1 - identifier و 2 - identifier ... هي أسماء لحقول موصوفة كدليل بعبارة
USAGE IS INDEX تستخدم لأجل حفظ قيمة الدليل أو إعادة القيمة إليه .

(٥) من معاني كلمة SET باللغة الانكليزية : يضع ، يحدد ، يعين ، يهيئ - المورد ١٩٨٢

أمثلة :

لإعطاء القيمة 1 للدليل :
 SET XD1 TO 1
 لحفظ قيمة الدليل في الحقل A1 :
 SET A1 TO XD1
 حيث :
 77 A1 USAGE IS INDEX
 SET XD2 TO A1

لإعطاء الدليل XD2 القيمة المحتواة في الحقل A1 .

أما الصيغة الثانية لتعليمة SET فتأخذ الشكل التالي :

2)

SET index - name - 1 [index - name - 2]

$$\left\{ \begin{array}{l} \text{UP BY} \\ \text{DOWN BY} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier - 4} \\ \text{literal - 2} \end{array} \right\}$$

حيث أن identifier - 4 أي حقل موصوف في قسم البيانات يحتوي على قيمة عددية تستخدم لزيادة أو نقص قيمة الدليل index - name - 1 أو أكثر من دليل في نفس التعليمة.

ويستفاد من هذا الشكل في تعديل قيمة الدليل زيادة أو نقصاناً من أجل الوصول الى الموضع اللازم للعنصر المطلوب في الجدول .

أمثلة :

لزيادة قيمة الدليل XD3 بالقيمة 1 :
SET XD3 UP BY 1.

ولزيادة قيمة الدليل XD2 بالعدد 2 :
SET XD2 UP BY 2.

ولانقاص قيمة الدليل XD1 بالقيمة الموجودة في الحقل INCR :

SET XD1 DOWN BY INCR.

*PERFORM VARYING تعليمة أنجز المتبدلة

إن تعليمة PERFORM مع عبارة VARYING من التعليمات المتقدمة في لغة كوبرول وهي مناسبة بشكل خاص عند استخدام الجداول في توصيف البيانات، وتصلح سواء كان الجدول موصوفاً مع دليل Indexed أو بدون دليل أي مع مؤشر Subscript. كما يمكن أن تكون ذات فائدة أكبر كلما كبر حجم الجدول ويمكن استخدامها مع البعد الواحد أو البعدين أو ثلاثة أبعاد.

وتأخذ تعليمة «أنجز» PERFORM مع VARYING الشكل العام التالي :

PERFORM procedure - name - 1 [THRU procedure - name - 2]

$$\begin{array}{c}
 \text{VARYING} \left\{ \begin{array}{l} \text{identifier - 1} \\ \text{index - name - 1} \end{array} \right\} \text{ FROM } \left\{ \begin{array}{l} \text{identifier - 2} \\ \text{index - name - 2} \\ \text{literal - 1} \end{array} \right\} \\
 \text{BY } \left\{ \begin{array}{l} \text{identifier - 3} \\ \text{literal - 2} \end{array} \right\} \text{ UNTIL Condition - 1} \\
 \left[\text{AFTER } \left\{ \begin{array}{l} \text{identifier - 4} \\ \text{index - name - 3} \end{array} \right\} \text{ FROM } \left\{ \begin{array}{l} \text{identifier - 5} \\ \text{index - name - 4} \\ \text{literal - 3} \end{array} \right\} \right. \\
 \left. \text{BY } \left\{ \begin{array}{l} \text{identifier - 6} \\ \text{literal - 4} \end{array} \right\} \text{ UNTIL Condition - 2} \right]
 \end{array}$$

(*) ADVANCED COBOL PP. 401 and
Pragmatic COBOL PP. 172 - 173

$$\left[\begin{array}{c} \text{AFTER} \left\{ \begin{array}{l} \text{identifier - 7} \\ \text{index - name - 5} \end{array} \right\} \text{FROM} \left\{ \begin{array}{l} \text{identifier - 8} \\ \text{index - name - 6} \\ \text{literal - 5} \end{array} \right\} \text{BY} \\ \left\{ \begin{array}{l} \text{identifier - 9} \\ \text{literal - 6} \end{array} \right\} \text{UNTIL Condition - 3} \end{array} \right]$$

حيث أنه يقصد بالشروط 1 - Condition و 2 - Condition أو 3 - Condition

عمليات المقارنة الوارد ذكرها مع تعليمة IF.

ولتوضيح كيفية استخدامها نأخذ المثال التالي :

لدينا جدول يحتوي على ثلاثة معاهد وكل معهد يشمل خمس مدارس وكل مدرسة لها نظام أربع سنوات . وفي كل سنة يوجد عدد من الطلاب . وموصوف هذا الجدول على النحو التالي :

1# ENROLLMENTS.

5# COLLEGE OCCURS 3 TIMES.

1# SCHOOL OCCURS 5 TIMES.

15 YEAR - S OCCURS 4 TIMES.

2# STUDENTS PIC 9 (4).

وعند جمع عدد جميع الطلاب في الجدول يمكن استخدام تعليمة PERFORM وفق الشكل التالي :

MOVE ZERO TO TOTAL - STUDENTS.

PERFORM SUMMATION VARYING I FROM 1 BY 1 UNTIL I > 3

AFTER J FROM 1 BY 1 UNTIL J > 5

AFTER K FROM 1 BY 1 UNTIL K > 4.

-٢١٧-

حيث أن فقرة SUMMATION تعادل الشكل :

SUMMATION.

ADD STUDENTS (I, J, K)

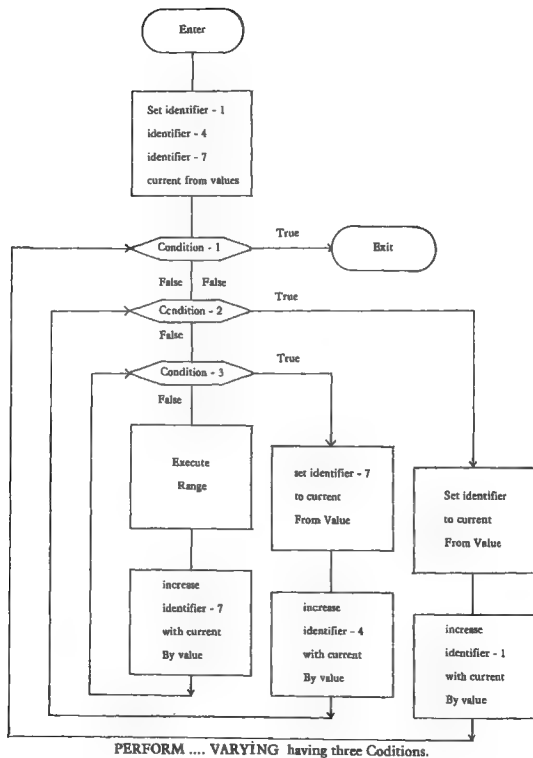
TO TOTAL - STUDENTS.

وأن I, J, K أسماء للمؤشرات الخاصة بالجدول ، كما يمكن أن تكون أسماء للأدلة Indices إذا كان الجدول موصوفاً مع جملة INDEXED BY .

و يلاحظ أن إعطاء القيمة الأولية للمؤشرات أو الأدلة يتم ضمن التعليمية وكذلك المقارنة على نهاية (حجم) الجدول ولكل بُعد من أبعاده .

كما يجب التنويه إلى أن جميع المؤشرات والأدلة يجب أن تكون قيمتها أكبر من الصفر.

والصفحة التالية توضح المخطط الانسيابي لكيفية تنفيذ تعليمية PERFORM مع VARYING بوجود ثلاثة شروط .



٦ - ٧ تعليمية البحث في الجدول SEARCH

من أجل تسهيل عملية البحث في الجدول عن عنصر محدد فقد اشتملت لغة كوبرول على تعليمية خاصة تسمى SEARCH ويمكن أن تأخذ هذه التعليمية الصيغتين العامتين * :

1)
$$\text{SEARCH identifier - 1} \left[\text{VARYING} \left\{ \begin{array}{l} \text{index - name - 1} \\ \text{identifier - 2} \end{array} \right\} \right]$$

[AT END imperative - statement - 1]

$$\text{WHEN Condition - 1} \left\{ \begin{array}{l} \text{imperative - statement - 2} \\ \text{NEXT SENTENCE} \end{array} \right\}$$

$$\left[\text{WHEN Condition - 2} \left\{ \begin{array}{l} \text{imperative - statement - 3} \\ \text{NEXT SENTENCE} \end{array} \right\} \right] \dots$$

2)
$$\text{SEARCH ALL identifier - 1}$$

[AT END imperative - statement - 1]

$$\text{WHEN Condition - 1} \left\{ \begin{array}{l} \text{imperative - statement - 2} \\ \text{NEXT SENTENCE} \end{array} \right\}$$

وفي كلا الصيغتين يشترط أن يكون الجدول محل البحث موصوفاً بعبارة OCCURS والتي تحتوي على جملة INDEXED BY .

وتتعامل تعليمة SEARCH مع الجدول على بُعد واحد فقط وإذا كان الجدول له أكثر من بُعد فإن البحث يفترض أن يتم على آخر عبارة OCCURS المذكورة في توصيف الجدول المتعدد الأبعاد بينما تبقى بقية الأبعاد ذات المستويات العليا ثابتة .

وجملة WHEN تعني وجود احتمالين لنتيجة البحث بواسطة التعليمة : إما الحصول على البند (السطر) المطلوب أو عدم الحصول عليه . والشرط الذي يلي كلمة WHEN (يسمى Condition) هو الذي يقرر البند اللازم والمطلوب في البحث . وعند تحقق الشرط ينتقل تسلسل التنفيذ الى التعليمة التالية لتعليمة SEARCH في البرنامج ما لم يذكر خلاف ذلك بدلا من جملة NEXT SENTENCE .

أما في حالة البحث عن كامل الجدول وعدم تحقق الشرط (أي عدم الحصول على البند المطلوب) ينتقل التسلسل في التنفيذ الى جملة AT END والتعليمة التالية لها مباشرة .

كما يمكن استخدام جملة NEXT SENTENCE بعد جملة AT END لنقل التسلسل الى التعليمة التالية لـ SEARCH مباشرة إذا رغب المبرمج في ذلك .

وإضافة جملة VARYING الى تعليمة البحث SEARCH تمكن المبرمج من استخدام دليل ثانوي غير الدليل الأول Prime ليكون عدداً في عملية اجراء البحث . وأحياناً يكون هذا الدليل الثانوي دليلاً في جدول آخر وإذا لم يكن معرّفاً بعبارة INDEXED BY فيجب أن يكون موصوفاً لحقل 2 - identifier بعبارة . USAGE IS INDEX

أما الحقل المسمى في الصيغة العامة : 1 - identifier وهو مجال البحث فهو عبارة عن اسم البند Item أو Entry الأساسي في الجدول والذي وصف مع عبارة OCCURS وجملة INDEXED BY . ولا يجوز استخدام البحث على غيره من حقول الجدول وخاصة تلك التي لم يرد وصفها في عبارة OCCURS .

٦-٧-١ البحث التسلسلي SEQUENTIAL SEARCH

تقوم تعليمية SEARCH بإجراء البحث على التتالي عندما لا تستخدم معها كلمة ALL، وعلى المبرمج أن يحدد نقطة بداية البحث في الجدول بإعطاء القيمة الأولية للدليل الخاصة بالبند محل البحث. وفي عملية البحث تزداد قيمة الدليل بالقيمة (1) آلياً ليتم الانتقال الى البند التالي وهكذا حتى يتحقق شرط الحصول في كلمة WHEN أو الوصول الى نهاية الجدول. ويتم المقارنة على نهاية البحث (قيمة الدليل العظمى) آلياً أيضاً ضمن هذه التعليمية.

والشكل التالي يبين مثالاً عن البحث التسلسلي :

البحث التسلسلي بتعليمية SEARCH

| | |
|----------|--|
| 9.15.45 | DEC 25, 1982 |
| 001506/ | |
| 001507 | 01 TABLE - AREA. |
| 001508 | 05 QQ - Y OCCURS 10 TIMES INDEXED BY XY. |
| 001509 | 10 QQ - Z OCCURS 10 TIMES INDEXED BY XZ, |
| 001510 | X - ZZ. |
| 001511 | 15 QQ - Z - 1 PIC X. |
| 001512 | 15 QQ - Z - 2 PIC S9999. |
| 001513 | 15 QQ - Z - 3 PIC S9999 COMP. |
| 001514 * | ----- |
| 001515 | 01 LOOK - FOR PIC S9 VALUE + 3. |
| 001516 | 01 MAX - Y PIC S9 (3) VALUE + 10. |
| 001517 | 01 MAX - Z PIC S9 (3) VALUE + 10. |
| 001518 | 01 CODE - IS PIC X. |

```

001519 * -----
001910 *
001920 P - INIT.
001930     PERFORM LOAD - ITEM VARYING XY FROM 1 BY 1
001940     UNTIL XY IS > MAX - Y.
001950 * .....
001960 * .....
001970     PERFORM FIND - ITEM VARYING XY FROM 1 BY 1
001980     UNTIL XY IS > MAX - Y.
001990     DISPLAY CODE - IS UPON CONSOLE.
001991     GO TO NXT - SEC.
001992 *
001993 FIND - ITEM SECTION.
001994 F - I - 1.
001995     SET XZ TO 1.
001996 *
001997     SEARCH QQ - Z
001998     AT END MOVE 'N' TO CODE - IS
001999     WHEN QQ - Z - 2 (XY, XZ) IS = LOOK - FOR
002000     AND QQ - Z - 3 (XY XZ) IS NOT = ZERO
002001     MOVE QQ - Z - 1 (XY XZ) TO CODE - IS
002002     SET XY TO 11
002003     SET XZ TO 11.
002004 *
002005 LOAD - ITEM SECTION.

```

002006 L - I - 1.

002007 PERFORM L - I - 2 VARYING XZ FROM 1 BY 1

002008 UNTIL XZ IS MAX - Z.

002009 GO TO L - I - EXIT.

002010 L - I - 2.

002011 MOVE 'A' TO QQ - Z - I (XY, XZ)

002012 SET QQ - Z - 2 (XY, XZ) TO XY.

002013 SET QQ - Z - 3 (XY, XZ) TO XZ.

002014 L - I - EXIT. EXIT.

002015 NXT - SEC SECTION.

002016 N - S - ALLL.

٦-٧-٢ البحث الثنائي BINARY SEARCH

البحث الثنائي يعتبر طريقة فعالة وسريعة في الجدول عن البند المطلوب و يتم في تعليمية البحث SEARCH آلياً مع إضافة كلمة ALL أي استخدام الصيغة رقم (٧) من الشكل العام السابق ذكره. كما يشترط لتحقيقه إضافة جملة بعبارة OCCURS عند توصيف الجدول تدل على طريقة ترتيب هذا الجدول وفق المفتاح الذي يتضمنه تصاعدياً أو تنازلياً لأنها سوف تحدد مسار البحث الثنائي في الجدول. وتصبح عبارة OCCURS عندئذ على النحو التالي في شكلها العام.

(level - number) (identifier) OCCURS (integer) TIMES

$$\left[\begin{array}{c} \underline{\text{ASCENDING}} \\ \underline{\text{DESCENDING}} \end{array} \right] \text{ KEY IS data - name - 1 } \left[\text{data - name - 2} \right] \dots$$

[INDEXED BY index - name - 1 [index - name - 2]]

وتجدر الإشارة الى أنه عند استخدام البحث الثنائي مع كلمة ALL لا توجد حاجة الى تحديد نقطة بداية البحث في الجدول بتعليمة SET للدليل وإنما تتم ألياً ضمن تعليمة SEARCH ذاتها. والشكل التالي يوضح مثالا عن استخدام هذه التعليمة.

**البحث باستخدام ALL بتعليمة SEARCH
(البحث الثنائي (BINARY**

| | 9.15.45 | DEC 25, 1982 |
|----------|--|--------------|
| 001530/ | | |
| 001540 | 01 TT - X. | |
| 001550 | 05 TT - Y OCCURS 5 TIMES | |
| 001560 | ASCENDING KEY IS TT - Y - 2 | |
| 001570 | INDEXED BY TT - NDX. | |
| 001580 | 15 TT - Y - 1 PIC X. | |
| 001590 | 15 TT - Y - 2 PIC S9999. | |
| 001700 * | | |
| 002019 | PERFORM LOAD - IT VARYING TT - NDX FROM 1 BY 1 | |
| 002020 | UNTIL TT - NDX IS > 5. | |
| 002021 | GO TO FIG - SEC. | |
| 002022 * | | |
| 002027 | LOAD - IT SECTION. | |
| 002028 | LD - 1. | |
| 002029 | MOVE 'X' TO TT - Y - 1 (TT - NDX). | |
| 002030 | SET TT - Y - 2 (TT - NDX) TO TT - NDX. | |
| 002031 | FIG - SEC SECTION. | |
| 002032 | FIG - 1. | |

| | |
|----------|---|
| 002034 | SEARCH ALL TT - Y |
| 002035 | AT END GO TO ERR - NO - TT |
| 002036 | WHEN TT - Y - 2 (TT - NDX) IS = 4 |
| 002039 | GO TO GOOD - TT - FOUND. |
| 002040 | ERR - NO - TT. |
| 002041 | DISPLAY 'NO ITEM FOUND ON TT - X TABLE' |
| 002042 | UPON CONSOLE. |
| 002043 | GO TO E - SEARCH - ALL. |
| 002044 * | |
| 002045 | GOOD - TT - FOUND. |
| 002046 | DISPLAY 'ITEM ON TT - X TABLE IS FOUND' |
| 002047 | UPON CONSOLE. |
| 002048 | E - SEARCH - ALL. |
| 002049 * | |
| 002050 * | |

أسئلة وتمارين

سؤال ١ - لدينا الجدول المبين فيما يلي والذي يدعى SALARY - TAB والمطلوب توصيفه في قسم البيانات باستخدام عبارة OCCURS

| SALARY - DATA (1) | | SALARY - DATA (2) | | SALARY - DATA (3) | |
|----------------------|----------|----------------------|----------|----------------------|----------|
| SALARY (1) | DATE (1) | SALARY (2) | DATE (2) | SALARY (3) | DATE (3) |
| | | | | | |
| | | | | | |

سؤال ٢ - لدينا الجدول التالي واسمه SAL - TAB والمطلوب توصيفه في قسم البيانات باستخدام عبارة (أو عبارات) OCCURS مع جملة INDEXED BY.

| SALARY - DATA | | | | | | | | | | | | | | | | | |
|---------------|--|--|------------|--|--|------------|--|--|----------|--|--|----------|--|--|----------|--|--|
| SALARY (1) | | | SALARY (2) | | | SALARY (3) | | | DATE (1) | | | DATE (2) | | | DATE (3) | | |
| | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |

سؤال ٣ - في التوصيفين التاليين جدولين يبين حجم الجدول لكل منهما بعدد البايت ؟ ثم وضح أي من التوصيفين يعتبر أكثر كفاية لأجل تداول الجدول.

1) 1 STATE - TABLE.

5 STATE - NAME OCCURS 50 TIMES PIC A (15).

5 STATE - PPLATION OCCURS 5 TIMES PIC 9 (8).

2) 1 STATE - TABLE.

5 NAME - PPLATION OCCURS 50 TIMES INDEXED BY XST.

1 STATE - NAME PIC A (15).

10 STATE - PPLATION PIC 9 (8).

حجم الجدول (1) بايت

حجم الجدول (2) بايت

الكفاية في التداول تعتبر أكبر: في التوصيف (1)

في التوصيف (2)

سؤال ٤ - لدينا الجزء التالي من برنامج يحتوي على توصيف لجدول باسماء المدن في المملكة وهناك تعليمة بحث SEARCH في قسم الاجراءات من أجل تحديد اسم المدينة بعد معرفة رمزها من ملف القراءة.

01 TABLE - AREA.

| | | | | | |
|----|--------|-----|--------|-------|------------------|
| 03 | FILLER | PIC | X (15) | VALUE | '01 RIYADH '. |
| 03 | FILLER | PIC | X (15) | VALUE | '02 JEDDAH '. |
| 03 | FILLER | PIC | X (15) | VALUE | '03 DAMMAM '. |
| 03 | FILLER | PIC | X (15) | VALUE | '04 QUASEEM'. |
| 03 | FILLER | PIC | X (15) | VALUE | '05 HAYEL '. |
| 03 | FILLER | PIC | X (15) | VALUE | '06 ALMADINA '. |
| 03 | FILLER | PIC | X (15) | VALUE | '07 ABHA'. |
| 03 | FILLER | PIC | X (15) | VALUE | '08 BALGURASHI'. |
| 03 | FILLER | PIC | X (15) | VALUE | '09 ASEEF'. |
| 03 | FILLER | PIC | X (15) | VALUE | '10 SHAQRA'. |

01 RED - TABLE - AREA REDEFINES TABLE - AREA.

05 LOCATIONS OCCURS 10 TIMES INDEXED BY LOC - INDEX.

| | | | |
|----|------------|-----|---------|
| 10 | LOC - CODE | PIC | 99. |
| 10 | LOC - NAME | PIC | X (13). |

PROCEDURE DIVISION.

.....

.....

SET LOC - INDEX TO 1.

SEARCH LOCATIONS AT END

MOVE 'UNKOWN' TO W - LOCATION

WHEN R - LOCATION = LOC - CODE (LOC - INDEX)

MOVE LOC - NAME (LOC - INDEX) TO W - LOCATION.

MOVE W - LOCATION TO P - LOCATION.

والمطلوب :

- بيان امكانية استخدام تعليمة البحث SEARCH مع كلمة ALL الاختيارية على هذا الجدول ؟ ولماذا اذا كان الجواب بالنفي ؟
- إعادة كتابة هذا الجزء بدون استخدام تعليمة SEARCH بحيث يؤدي نفس الغرض.

سؤال ٥ - لدينا توصيف الجدول في قسم البيانات كما يلي :

1) 01 PAYMENTS

05 FILLER PIC X (8) VALUE "17161819".

05 FILLER PIC X (8) VALUE "14561554".

05 FILLER PIC X (8) VALUE "12143748".

01 PAYMENTS - TAB REDEFINES PAYMENTS.

05 STARTING - PERIOD OCCURS 3 TIMES INDEXED BY
S - INDEX.

10 ENDING - PERIOD OCCURS 2 TIMES INDEXED BY
E - INDEX.

15 PAYMENT PIC 99V99.

وفي قسم الاجراءات التعليمات التالية :

2) SET S - INDEX TO 3.

SET E - INDEX TO 1.

MOVE PAYMENT (S - INDEX, E - INDEX) TO PAYMENT - OUT.
 SET S - INDEX DOWN BY 1.
 MOVE PAYMENT (S - INDEX, E - INDEX) TO PAYMENT - LIST.

والمطلوب :

- ١ - رسم توضيحي لشكل الجدول الموصوف PAYMENTS على شكل صفوف وأعمدة.
- ٢ - بيان قيمة الحقل PAYMENT - OUT وقيمة الحقل PAYMENT - LIST . بعد تنفيذ التعليمات المذكورة سابقاً في قسم الاجراءات علماً بأن شكل الحقلين هو 99.99 .

تمرين ١ - أكتب البرنامج التالي :

لدينا الجدول التالي والذي يمثل قيمة مبيعات إحدى الشركات لثلاث سلع مختلفة وعلى مدى أيام الأسبوع الستة . (المبالغ بالآلاف الريالات) :

| DATE | GOODS | | |
|-------|-------|---------|-------|
| | FOOD | CLOTHES | EQPMT |
| SAT | 10 | 20 | 30 |
| SUN | 02 | 03 | 04 |
| MON | 15 | 16 | 17 |
| TUE | 40 | 41 | 42 |
| WED | 08 | 09 | 10 |
| THR | 50 | 51 | 52 |
| TOTAL | ??? | ??? | ??? |

المطلوب :

- أ - توصيف هذا الجدول في قسم البيانات من أجل إجراء عمليات الجمع والطباعة .
- ب - كتابة الاجراءات اللازمة لجمع قيمة السلع GOODS الثلاث في أيام الاسبوع كما هو مبين في الجدول .
- ج - كتابة التعليمات والاجراءات اللازمة لطباعة هذا الجدول على الشكل التالي :

| DAY | FOOD | CLOTHES | EQUIPMENT |
|-------|------|---------|-----------|
| XXX | 999 | 999 | 999 |
| . | . | . | . |
| . | . | . | . |
| . | . | . | . |
| TOTAL | 999 | 999 | 999 |

ملاحظة : استخدم الفراغات وأسماء الحقول المناسبة واستعن برسم FLOWCHART

الفصل السابع

الفرز والدمج SORT / MERGE

٧-١ ملحة عامة

من أجل زيادة السرعة في معالجة الملفات التسلسلية Sequential التي تحتوي على حقل أو أكثر يعتبر بمثابة مفتاح Key للسجل الموجود في الملف ، أصبح من الواجب فرز (Sort) هذه الملفات طبقاً لتسلسل معين في هذا الحقل : أما تنازلي Descending أو تصاعدي Ascending .

وعملية الفرز يمكن أن تتم بأحدى الطرق الثلاث التالية :

١ - الفرز بواسطة إجراء معين يكتبه المبرمج في برنامجه أو في برنامج خاص وهناك طرق عدة واساليب مختلفة لإجراء الفرز لا مجال لذكرها هنا .

٢ - الفرز بواسطة برنامج جاهز خارج برنامج كوبرول يدعى أحياناً Sort Utility وهذا البرنامج الجاهز يختلف من شركة صانعة الى أخرى (من حاسب الى آخر) ويعطي المبرمج أو المشغل الفرصة بادخال المعطيات الخاصة Parameter بمتطلبات عملية الفرز مثل اسم الملف المراد فرزه وطول السجل واسم الحقل الذي يتم الفرز على أساسه (Major Key) واسماء الحقول الأخرى (Minor Keys) واسم الملف الذي يحتوي نتيجة الفرز ومعلومات أخرى تتعلق بعملية الفرز ولا داعي للدخول في تفاصيله لأنه يختلف باختلاف نوع الحاسب ولكن من خواصه أنه سهل التطبيق وسريع التنفيذ .

٣ - الفرز بواسطة تعليمات خاصة تذكر في داخل برنامج كوبول تؤدي الى تنفيذ عملية فرز الملف و ثم متابعة تنفيذ البرنامج الذي يحتوي تعليمات الفرز، وهذا الاسلوب الذي يهم المبرمج بلغة كوبول معرفته في هذا المؤلف .

٧-٢ الفرز في لغة كوبول SORT

بما أن عملية الفرز مطلوبة غالباً للمحافظة على نظام تسلسلي للبيانات فإن لغة كوبول قد طورت لتشمل (بعد عام ١٩٧٤) ميزة الفرز هذه وتوفر على المبرمج في الدخول في تفاصيل الطرق التقنية للفرز ولتكون بعيدة عن التأثير بنوع الحاسب المستخدم ومستقلة عن برامجها الجاهزة . ولتوضيح عملية الفرز نأخذ المثال التالي :

لنفترض أنه يوجد لدينا ملف على شريط ممغنط (أو على قرص ممغنط رتب على أساس تسلسلي) يحتوي على وصف التالي في قسم البيانات Data Division :

01 INTAPE - REC.

05 STDNT - NO PIC 9 (7).

05 STDNT - NAME PIC X (30).

05 STDNT - MRKS.

07 SCORE PIC 99V99.

07 GRADE PIC X.

05 FILLER PIC X (38).

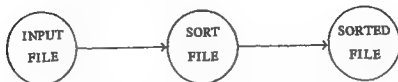
ونرغب في فرز هذا الملف تصاعدياً وفق رقم الطالب STDNT - NO وتنازلياً وفق العلامة SCORE في آن واحد . ونفترض بأن الملف المفروز يدعى SORTED - FILE

وسوف يحفظ على وحدة تداول ثالثة . فيمكن تصور عملية الفرز والتي سوف تشتمل على ثلاثة ملفات :

INPUT - FILE

و SORT - FILE

و SORTED - FILE



لأن عملية الفرز تحتاج الى ملف (احياناً أكثر من ملف) ليستخدام بمثابة منطقة عمل خاصة بالفرز (SORT FILE).

والشكل المبين في الصفحة التالية يمثل برنامج كويول يحتوي على عملية فرز للملف الموصوف في الفقرات السابقة والذي يحتوي على معلومات عن الطلاب ويتم الفرز وفق رقم الطالب تصاعدياً ووفق علامته تنازلياً .

و يلاحظ في الشكل أنه تم توصيف ثلاثة ملفات في قسم المحيطات ENVIRONMENT DIVISION بعبارة SELECT وكذلك في قسم توصيف البيانات DATA DIVISION ثم توصيف ملف الادخال INPUT - FILE وملف الفرز SORT - FILE وملف الاخراج SORTED - FILE . وتستخدم عبارة SD لملف الفرز في التوصيف و يقصد بها Sort file Description بدلاً من FD المستخدمة في الملفات الأخرى . كما يلاحظ في هذا القسم عدم وجود عبارة ... LABEL RECORD لملف الفرز (رغم وجود بعض المترجمات التي تسمح بها) لأن ANSI COBOL اعتبرت ملف الفرز كمنطقة تخزين وعمل داخلي في البرنامج .

IDENTIFICATION DIVISION.

PROGRAM-ID. SORTP.

AUTHOR. AJND-KHAYAT.

** THIS PROGRAM IS TO ILLUSTRATE **

** COBOL SORT STATEMENT. **

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER. IBM-3033.

OBJECT-COMPUTER. IBM-3033.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT INPUT-FILE ASSIGN TO

UT-S-SYSIN.

SELECT SORT-WORK-FILE ASSIGN TO

UT-S-WORKSORT.

SELECT SORTED-FILE ASSIGN TO

UT-S-SORTOUT.

DATA DIVISION.

FILE SECTION.

FD INPUT-FILE LABEL RECORD IS STANDARD

DATA RECORD IS INPUT-REC.

01 INPUT-REC.

05 STD-NO PIC 9(7).

05 STD-NAME PIC X(30).

05 STD-MARK.

07 STD-SCORE PIC 99V99.

07 STD-GRADE PIC X.

05 FILLER PIC X(38).

*

SD SORT-WORK-FILE

RECORD CONTAINS 80 CHARACTERS

DATA RECORD IS SORT-REC.

01 SORT-REC.

05 STD-NO-S PIC 9(7).

05 STD-NAME-S PIC X(30).

05 STD-MARK-S.

07 STD-SCORE-S PIC 99V99.

07 STD-GRADE-S PIC X.

05 FILLER PIC X(38).

FD SORTED-FILE

LABEL RECORD IS STANDARD

```

                                DATA RECORD IS SORTED-REC.
01  SORTED-REC                      PIC X(80).
*
PROCEDURE DIVISION.
SORT-ROUTINE.
    SORT  SORT-WORK-FILE
        ON  ASCENDING KEY STD-NAME-S
        ON  DESCENDING KEY STD-SCORE-S
    USING  INPUT-FILE
    GIVING SORTED-FILE.
STOP RUN.

```

```

**
***  UNSORTED  FILE                      *****
0234517AHAMD                          7050C
0234527ZAID                           8050B
0234537ALI                             9050A
0234547WALEED                         5050F
0234557FAROUQ                        6050D
0234557HISHAM                        6050D
0234557JAMEEL                        6650D
0234557ABED-ALAZIZ                   9650A
0234557MAZEN                         8650A
0234557NABEEL                        7650A
**
***  SORTED  FILE                      *****
0234557ABED-ALAZIZ                   9650A
0234517AHAMD                          7050C
0234537ALI                             9050A
0234557FAROUQ                        6050D
0234557HISHAM                        6050D
0234557JAMEEL                        6650D
0234557MAZEN                         8650A
0234557NABEEL                        7650A
0234547WALEED                         5050F
0234527ZAID                           8050B

```

أما قسم الإجراءات PROCEDURE DIVISION فهو بسيط و يشتمل على فقرة واحدة، وهو أمر الفرز SORT وهذا الأمر (أو التعليمة) فعال ومريح للمبرمج حيث أنه لا يحتاج الى توصيف معقد لأدلة الفرز Keys ومصدر البيانات (الادخال) ووجهة البيانات (الايخراج)، بل و يُكفى بذكر بسيط وسهل لهذه المعطيات.

وفي حال كون المبرمج بحاجة الى عملية الفرز على أكثر من دليل هـ (مفتاح) فيمكنه إضافة اسم المفتاح بعد كلمة ASCENDING أو DESCENDING و يعتبر الاسم الأول بمثابة مفتاح كبير Major Key والتالي له مفتاح أصغر يدعى عادة Minor Key. وعدد هذه المفاتيح في التعليمة يقارب سبعة مفاتيح ويختلف من مترجم كومبول الى آخر وامكانية نظام التشغيل المتوفرة.

وعبارة USING INPUT - FILE تحدد الملف الذي يمثل مصدر البيانات المراد فرزها بينما عبارة GIVING SORTED - FILE تحدد وجهة البيانات بعد الفرز.

و يلاحظ في الشكل السابق أيضاً أن عملية فتح وإغلاق OPEN ، CLOSE للملفات لم تذكر صراحة في البرنامج لأن تعليمة SORT تتضمن ذلك آلياً.

الشكل العام لتعليمة الفرز SORT

فيما يلي عرض للأشكال العامة للتوصيفات اللازمة لتعليمة الفرز في أقسام البرنامج المختلفة.

ففي قسم التعريف IDENTIFICATION لا يوجد أي استثناء عن التوصيف العادي.

(هـ) دليل أو مفتاح الفرز Keys

أما في قسم المحيطيات ENVIRONMENT وفي الجزء الخاص بالملفات تأخذ عبارة
SELECT الشكل التالي :

FILE - CONTROL.

SELECT file - name ASSIGN TO device - name - 1

[device - name - 2] [OR device - name - 3]

[device - name - 4] ...

[FOR MULTIPLE { REEL
UNIT }]

ووجود كلمة OR في هذه العبارة تسمح باخراج نتيجة الفرز على احدى الوسائط
(وحدات تداول التخزين الخارجى) المعرفة بعد هذه الكلمة .

أما قسم البيانات DATA فيمكن أن يظهر على الشكل التالي :

SD file - name

[RECORD CONTAINS [integer - 1 TO] integer - 2 CHARACTERS] ,

[DATA { RECORD IS
RECORDS ARE } data - name - 1
[data - name - 2] ...]

حيث أن كلمة SD تدل على وصف لملف الفرز والعبارة التالية
RECORD CONTAINS تحدد طول السجل RECORD الذي يحتوي ملف الفرز
وحسب بالبايت Bytes . و يلاحظ هنا أن عبارة BLOCK CONTAINS الاختيارية لم
تذكر لأن حجم الشكل Blocks سوف يقوم روتين الفرز بتحديد آلياً .

أما عبارة DATA و RECORD أو RECORDS فتصف اسم (أو أسماء) السجل (السجلات) الموصوف (الموصوفة) تحت اسم ملف الفرز. أما قسم الإجراءات PROCEDURE فيمكن أن يحتوي تعليمة الفرز التي تأخذ الشكل العام التالي مع عباراتها الاختيارية :

SORT file - name - 1

ON $\left\{ \begin{array}{l} \underline{\text{ASCENDING}} \\ \underline{\text{DESCENDING}} \end{array} \right\}$ KEY data - name - 1 [, data - name - 2]

$\left[\text{ON} \left\{ \begin{array}{l} \underline{\text{ASCENDING}} \\ \underline{\text{DESCENDING}} \end{array} \right\} \text{KEY data - name - 3 [, data - name - 4] ...} \right]$

[COLLATING SEQUENCE IS alphabet - name]

$\left\{ \begin{array}{l} \underline{\text{INPUT PROCEDURE IS}} \text{ Section - name - 1} \left[\begin{array}{l} \underline{\text{THRU}} \\ \underline{\text{THROUGH}} \end{array} \right] \text{Sect - name - 2} \\ \underline{\text{USING}} \text{ file - name - 2 [, file - name - 3] ...} \end{array} \right\}$

$\left\{ \begin{array}{l} \underline{\text{OUTPUT PROCEDURE IS}} \text{ Section - name - 3} \left[\begin{array}{l} \underline{\text{THRU}} \\ \underline{\text{THROUGH}} \end{array} \right] \text{Sect - name - 4} \\ \underline{\text{GIVING}} \text{ file - name - 4} \end{array} \right\}$

حيث أن 1 - file - name بعد فعل الفرز SORT يقصد به ملف الفرز الموصوف في عبارة SD. كما أن 2 - data - name و 3 - data - name و 4 - data - name هي أسماء الحقول المعرفة في ملف الفرز والتي تعتبر بمثابة مفاتيح فرز. أما كلمة

ASCENDING تستخدم للفرز تصاعدياً و DESCENDING للفرز تنازلياً بالنسبة
لاسم المفتاح الذي يليهما . فمثلاً :

SORT file - name ON ASCENDING KEY

YEAR, STDNT - NAME, STDNT - NO

وهنا يتم الفرز تصاعدياً وفق السنة YEAR ثم اسم الطالب STDNT - NAME ثم رقم
الطالب STDNT - NO داخل الاسم وداخل السنة . وهذا الشكل يعادل :

SORT file - name ON ASCENDING KEY YEAR

ON ASCENDING KEY STDNT - NAME

ON ASCENDING KEY STDNT - NO

وعبارتنا INPUT PROCEDURE و OUTPUT PROCEDURE تشيران الى اسم
القسم (Section) أو مجموعة الفصول ٥ عند استخدام كلمة THRU.

والتعليمات التي يمكن أن ترد في فصل أو فصول عبارة INPUT PROCEDURE
تقوم بإجراء بعض المعالجة على الحقول أو السجلات الواردة من ملف الإدخال مثل
كتابتها على ملف الفرز لأجل فرزها ، وغالباً ما تكون هذه المعالجة مؤقتة ونحويلية . وفي
هذه الحالة يجب على المبرمج القيام بفتح الملف (ملف الإدخال) واستعمال تعليمة
القرأة (READ) و ثم معالجته . ولأجل الكتابة على ملف الفرز تستخدم تعليمة
RELEASE والتي تعادل تعليمة WRITE الوارد ذكرها في الفصول السابقة . وتأخذ
تعليمة RELEASE الشكل العام التالي :

RELEASE record - name [FROM identifier]

(٥) الفصل هو مجموعة فقرات من قسم الإجراءات

حيث يبدل record - name على اسم سجل ملف الفرز. وتساعد هذه التعليمة في نقل محتويات ملف الإدخال الى ملف الفرز.

وكذلك الحال في عبارة OUTPUT PROCEDURE التي تحتوي تعليمات خاصة بمعالجة تحويلية أو مؤقتة للسجلات بعد قراءتها من ملف الفرز وإرسالها الى ملف الإخراج. وتستخدم هنا تعليمة خاصة بالقراءة من ملف الفرز RETURN وتأخذ الشكل العام التالي :

RETURN file - name RECORD [INTO identifier]

AT END imperative statement

حيث تمثل عبارة file - name اسم ملف الفرز. وعند استخدام عبارة OUTPUT PROCEDURE يجب على المبرمج مراعاة فتح ملف الإخراج وإغلاقه.

أما عبارتا USING و GIVING فهما تستخدمان عندما لا توجد إجراءات خاصة عند ادخال البيانات الى ملف الفرز أو عند إخراجها منه. ويأتي اسم ملف الإدخال بعد كلمة USING وأحياناً يكون أكثر من ملف. أما اسم ملف الإخراج (أو وجهة البيانات بعد الفرز) فيذكر بعد كلمة GIVING.

وأخيراً تعطي عبارة COLLATING SEQUENCE المبرمج فرصة لتحديد تسلسل (ترتيب) أبجدي معين للبيانات المراد فرزها، وقد يختلف هذا الترتيب عن الترتيب القياسي المعروف باسم EBCDIC أو باسم ASCII ويحدد اسم الترتيب تحت اسم الفقرة الخاصة التالية :

SPECIAL - NAMES

في قسم المحيطيات.

٧-٣ دمج الملفات FILE MERGING

إن دمج الملفات إجراء شائع في معالجة البيانات والدمج يعني ببساطة جمع ملفين أو أكثر في ملف واحد. فمثلاً إذا كان هناك مصرف يوجد فيه ملف على شريط ممغنط للعمليات اليومية مرتبة حسب رقم الزبون وبشكل يومي عندئذ تتطلب المعالجة جمع هذه الملفات اليومية (المفرزة وفق رقم الزبون) في ملف واحد شهري بحيث تظهر كافة العمليات مرتبة مع بعضها لكل زبون أو عميل.

ولأن المبرمج يحتاج إليه تكراراً ولأنه ذو أهمية كبيرة فقد طبق في لغة كوبول باعتبارها ميزة في لغة عالية جداً، وأخذ هذا التطبيق شكل تعليمة تسمى الدمج MERGE.

ولنأخذ مثال على ذلك أن شركة أعمال تنتج (تخرج) في نهاية كل ربع سنة مواداً وتم تسجيل بيانات هذه المواد على ملف وكل سجل في هذا الملف يحتوي على رقم القسم ورقم المادة المباعة بالإضافة الى بيانات أخرى. وهذا الملف مُرَز (رُتَّب) حسب رقم القسم كمفتاح أساسي Major Key ورقم المادة (المنتج) المباعة كمفتاح فرعي Minor في عملية الفرز. وفي نهاية العام الشركة مهتمة بدمج هذه الملفات الأربعة الربعية وجمعها في ملف واحد سنوي.

والشكل التالي يوضح الاطار العام للاجزاء اللازمة في برنامج كوبول :

| | | |
|----|----------------|----------------------------|
| FD | FIRST - QRTR | LABEL RECORD STANDARD |
| | | DATA RECORD SALES - HSTRY. |
| 01 | SALES - HSTRY. | |
| 05 | DEPT - NO | PIC 99. |

| | | | | |
|-----------|----|---|--------------------------|--------|
| | 05 | PRODUCT - NO | PIC | 9 (6). |
| FD | 01 | SECOND - QRTR ... | | |
| FD | 01 | THIRD - QRTR ... | | |
| FD | 01 | FOURTH - QRTR ... | | |
| FD | 01 | YEARLY - FL | LABEL RECORD STANDARD | |
| | | | DATA RECORD YR - REC. | |
| 01 | | YR - REC. | | |
| | 05 | YDEPT - NO | PIC | 99. |
| | 05 | YPRODUCT - NO | PIC | 9 (6). |
| SD | | MERGE - FILE | DATA RECORD MERGE - REC. | |
| 01 | | MERGE - REC. | | |
| | 05 | MDEP - NO | PIC | 99. |
| | 05 | MPRD - NO | PIC | 9 (6). |
| PROCEDURE | | DIVISION. | | |
| MERGE | | MERGE - FILE ON ASCENDING KEY MDEP - NO | | |
| | | ON ASCENDING KEY MARD - NO | | |
| | | USING FIRST - QRTR, SECOND - QRTR, | | |
| | | THIRD - QRTR, FOURTH - QRTR | | |
| | | GIVING YEARLY - FL. | | |

يلاحظ من الشكل السابق أن هناك أربعة ملفات موصوفة بجملة FD ثم هناك ملف موصوف بجملة SD ليستخدم في عملية الدمج والذي اسمه MERGE - FILE .
و يلاحظ تطابق تفصيلات السجل في هذا الملف مع السجل في الملفات الأخرى .

وفي قسم الاجراءات تشير عبارة MERGE الى الملف الموصوف بـ SD والذي يحدد الدمج على أساس المفتاح الأساسي (رقم القسم) MDEP - NO والمفتاح الذي هو رقم المادة - NO - MPDR .

وجملة USING تحدد الملفات التي سوف تدمج والتي تعتبر ملفات إدخال . وهذه الملفات يجب أن تكون مغلقة CLOSED عند عملية الدمج لأن فتح الملف بوساطة MERGE يتم ضمناً .

وجملة GIVING تحدد الملف الناتج من عملية الدمج وهو الذي يحتوي على الملفات الأربعة الربع سنوية . وهذا الملف الجديد سوف يكون بنفس الترتيب SORT الذي كانت عليه الملفات الربع سنوية .

والشكل العام (الصيغة العامة) لتعليمة MERGE كما يلي :

MERGE file - name - 1

ON { ASCENDING
DESCENDING } KEY data - name - 1 [, data - name - 2] ...

[ON { ASCENDING
DESCENDING } KEY data - name - 3 [, data - name - 4]] ...

[COLLATING SEQUENCE IS alphabet - name]

USING file - name - 2, file - name - 3 [, file - name - 4]

[OUT PUT PROCEDURE IS section - name - 1 $\left[\begin{array}{c} \text{THROUGH} \\ \text{THRU} \end{array} \right.$ section - name - 2]
GIVING file - name - 5]

وعبارة OUTPUT PROCEDURE تعادل في استخدامها نفس العبارة المذكورة مع تعليمة SORT. وكذلك تستخدم عبارة RETURN في روتين إجراء الإخراج لجعل السجلات المدموجة جاهزة للمعالجة، تماماً كما في حالة الـ SORT. ولكن بعكس حالة SORT فإن عبارة MERGE لا تشمل أي اختيارات تتعلق بروتين إجراء الإدخال، لأن ملفات الإدخال يجب أن تكون في شكل ملائم ومناسب لعملية الدمج مثل تنفيذ عبارة MERGE.

٧ - ٤ اسئلة وتمارين

سؤال ١ - بين الصحيح والخاطئ من العبارات التالية :

- إذا ذكرت عبارة USING في تعليمة SORT فيجب ذكر عبارة GIVING معها.
- إذا تم ذكر عبارة INPUT PROCEDURE مع تعليمة SORT عندئذ يجب ذكر عبارة OUTPUT PROCEDURE
- يسمح بذكر مفتاح تصاعدي واحد أو مفتاح تنازلي واحد في تعليمة SORT.

- تستخدم تعليمة RELEASE في فقرات INPUT PROCEDURE .
- تستخدم تعليمة RETURN في فقرات OUTPUT PROCEDURE .
- إذا استخدمت عبارتا USING / GIVING مع فصل الفرز فإن الملف المفروز SORTED FILE يجب أن يحتوي على كل سجل من ملف الإدخال . INPUT FILE
- يوصف اسم ملف الفرز و ملف الدمج في قسم البيانات في فصل الملفات بعارة . SD
- لا تقبل تعليمة MERGE أربعة ملفات كملفات إدخال .
- تستخدم تعليمة MERGE لدمج ملف أو أكثر من ملف واحد .

سؤال ٢ - لديك أسماء الحقول التالية مع بياناتها :

| الاسم | الجهة | القسم |
|------------|--------|-------|
| محمد | الرياض | ١٠٠٠ |
| سامي | أبها | ٢٠٠٠ |
| ممدوح | أبها | ٢٠٠٠ |
| ياسر | جدة | ٣٠٠٠ |
| عبدالله | القصيم | ٤٠٠٠ |
| عبد الحميد | الرياض | ١٠٠٠ |
| عبد الرحمن | الرياض | ١٠٠٠ |
| بخيت | الدمام | ٥٠٠٠ |

اعد ترتيب البيانات السابقة وفق ما يلي :

- ١ - المفتاح الأولي MAJOR KEY هو حقل القسم (تنازلياً)
- ٢ - المفتاح الثانوي MINOR KEY هو حقل الاسم (تصاعدياً)

سؤال ٣ - في العبارة التالية :

sort sort - file

ascending key std - name

descending key year - in - school

ascending major

using file - one

giving file - two.

أجب على الأسئلة التالية :

- أ - ما هو المفتاح الأساسي؟
- ب - ما هو المفتاح الثانوي؟
- ج - أي من الملفات السابقة سيوصف تحت عبارة SD؟
- د - أي من الملفات سوف يوصف في عبارة SELECT؟
- هـ - أي ملف سيكون ملف إدخال؟
- و - أي من الملفات يجب أن يحتوي على أسماء الحقول year - in - school, std - name, و major؟

تمرين ١ :

ثلاثة ملفات تحوي على معلومات عن سجلات زبائن أحد المصارف موجودة على قرص ممسّط، المطلوب كتابة برنامج يجري العمليات التالية :

- ١ - دمج merge هذه الملفات الثلاثة في ملف واحد مفروز تصاعدياً على رقم الزبون ووضع الناتج على ملف اخراج اسمه OTMRG.
- ٢ - طباعة تقرير من ملف OTMRG يحوي على رقم الزبون واسمه والرصيد على أن

يطبع في نهاية العمل عدد الزبائن ومجموع الأرصدة.

٣- إذا كان الرصيد سالباً أطلع عبارة «*****» في حقل الملاحظات علماً بأن شكل السجل في الملفات الأربعة هو:

| | FIELD NAME | PICTURE |
|------------|-------------|------------|
| رقم الزبون | CUST - NO | 9 (6) |
| اسم الزبون | CUST - NAME | X (24) |
| الرصيد | CUST - BAL | S9 (8) V99 |
| غير مستعمل | CUST - DATA | X (40) |

• اختر اسماء الملفات والحقول المناسبة.

•• اطبع في الصفحة ٣٠ سطراً فقط. واترك سطر فراغ بين كل سطري طباعة - انظر

شكل توصيف المخرجات صفحة ٦٠٥.

••• أظهر الإشارة الجبرية في حقل الرصيد.

تمرين ٢ :

ملف يحتوي على سجلات سلع لاحدى المخازن اسمه STOCKF موجود على قرص

ممنقط، وشكل السجل فيه كالتالي :

| | FIELD NAME | PICTURE |
|------------|------------|-----------|
| رقم السلعة | STK - NO | 9 (5) |
| وصفها | STK - DESC | X (24) |
| الكمية | STK - QTY | 9 (5) |
| رمز السلعة | STK - CODE | 9 (3) |
| سعر الوحدة | STK - PRC | 9 (3) V99 |
| حد الطلب | STK - MIN | 9 (5) |
| غير مستعمل | FILLER | X (37) |

هذا الملف STOCKF ترتيب السلع فيه ترتيباً عشوائياً. المطلوب كتابة برنامج يجري العمليات التالية :

- ١ - فرز SORT هذا الملف تصاعدياً على رقم السلعة STK - NO وأن يوضع ناتج عملية الفرز على ملف آخر اسمه SRTSTK به نفس شكل وطول ملف الإدخال.
- ٢ - طباعة الملف SRTSTK على الطابعة بشكل تقرير يظهر رقم السلعة ، ووصفها ، والكمية ، ورمز السلعة ، وسعر الوحدة.
- ٣ - مقارنة الكمية STK - QTY مع حد الطلب STK - MIN فإذا كانت الكمية مساوية أو أصغر لحد الطلب اطبع عبارة أعد الطلب 'REORDER' في حقل الملاحظات - انظر شكل توصيف المخرجات صفحة ٦٠٦ .

ملاحظات :

- ١ - اطبع في الصفحة الواحدة ٢٥ سلعة فقط .
- ٢ - اطبع في رأس كل صفحة العناوين والتاريخ ورقم الصفحة .
- ٣ - اترك فراغاً بين كل سطري طباعة .

الفصل الثامن

الملفات FILES ومعالجتها

٨ - ١ ملحة عامة

نعلم إنه في أي نظام يدوي يتم تخزين البيانات DATA في ملفات ورقية - أو بلاستيكية - ويكتب عليها معلومات أساسية مثل محتوى هذا الملف، رقم الملف، تاريخه وإلى ما هنالك من معلومات تخص كل نظام.

أما عندما نريد تحويل النظام اليدوي CLERICAL SYSTEM الى نظام آلي ينفذ عن طريق الحاسب، فتكون بحاجة الى ملفات على الحاسب تخزن البيانات المختلفة للنظام. ويمكن معالجتها والرجوع إليها بسهولة باستخدام برامج الحاسب، هذه الملفات الخاصة بالحاسب لها أنواع وبنية وتنظيم وأساليب معالجة تختلف عن الملفات اليدوية - باختلاف وسط التخزين طبعاً - لتوفر السرعة في استدعاء البيانات والدقة في معالجة هذه البيانات، واختصار حجم التخزين طبعاً عن الملفات اليدوية.

٨ - ٢ أنواع الملفات *TYPE OF FILES

يمكن تقسيم أنواع الملفات حسب طبيعة البيانات التي تحتويها الى الأنواع التالية :

٨ - ٢ - ١ ملفات أساسية : MASTER FILES

وهذه من أهم أنواع الملفات والتي يركز عليها أي نظام، لما تحتويه من بيانات

أساسية، هذه البيانات تتكرر دوماً في أي دورة تشغيل للنظام. وهنالك نوعان من الملفات الأساسية :

٨-٢-١ ملفات مصدرية REFERENCE FILES :

وتحتوي على بيانات لا تتغير بصورة دورية مثل اسم الموظف، عنوانه، تاريخ ميلاده، رقم حفيظة النفوس (جواز سفره)، ... هذه المعلومات شبه ثابتة تقريباً.

٨-٢-٢ ملفات كثيرة التبدل DYNAMIC FILES :

وتحتوي على بيانات تتغير وفق احداث أو تعديلات عملية TRANSACTIONS معينة.

مثل ملف المخزون في نظام المستودعات. يظهر الكميات الجديدة المضافة الى المستودع، والكميات التي خرجت من المستودع والرصيد الحالي لكل صنف من أصناف المخزون.

كما أن ملف حجوزات السفر يظهر المسافرين الذين تم حجز أماكن لهم والوضع الحالي للأماكن الشاغرة.

ويمكننا أن نضيف الى أن النوعين السابقين من الملفات الأساسية المصدرية والمتبدلة يمكن أن يكونا منفصلين. ويمكن أن يجتمعا في ملف واحد مختلط يجمع بين مميزات الملفين، مثل نظام المحاسبة، فالسجل الواحد يمكن أن يجمع معلومات مصدرية مثل اسم الحساب، ورقم الحساب. ومعلومات خاصة بالحساب، كما يحتوي على المعلومات المالية المتغيرة. مثل الأرصدة لكل حساب.

٨-٢-٢ ملفات إدخال INPUT FILES :

تحمل هذه الملفات البيانات الى النظام من خلال تدقيقها ببرامج تحقق VALIDATION PROGRAM المكتوبة على الحاسب. اذ يتم ادخال البيانات عبر

بطاقات مثقبة PUNCHED CARDS أو عن طريق الشاشات TERMINALS الى هذه الملفات (تكون وقتها OUTPUT ملفات اخراج) . بعدها تعالج هذه الملفات كملفات إدخال للنظام ، وقرر على برنامج تحقق يسجل البيانات الصحيحة على ملفات أساسية والبيانات غير الصحيحة على ملفات أخرى أو يطبع بهم تقرير يظهر الخطأ ، ثم تصحح هذه الأخطاء وتتم معالجة الملفات من جديد .

٨-٢-٣ ملفات انتقالية : TRANSFER FILES

تحتوي هذه الملفات على بيانات تساعد في مراحل متقدمة من تشغيل النظام مثل ملفات الطباعة أو تحديث ملف آخر UPDATE ويمكن أن تكون مفروزة - SORTED - حسب مفتاح - KEY - معين .

٨-٢-٤ ملفات عمل : WORK FILES

هذه الملفات هي ملفات بسيطة تحتوي على بيانات مختارة من ملف أو أكثر . وهذه البيانات المختارة تحتاج لمعالجة مؤقتة ولا يهتم النظام لحفظها لفترة طويلة مثل ملفات SORT الفرز، أو الملفات التي يُحتاج إليها لإجراء بعض العمليات الحسابية الطويلة وتخزينها مؤقتاً على ملفات خاصة .

٨-٢-٥ ملفات إخراج : OUTPUT FILES

هذه الملفات تحمل البيانات من الحاسب الى الطباعة مثلاً أو لإدخالها في نظام آلي آخر .

مثال :

الفواتير التي سترسل الى الزبائن تطبع على الطباعة (ملف اخراج) أما الفواتير التي ستدخل في النظام المحاسبي الآلي فانها تسجل على ملف إخراج ، يكون فيما بعد ملف

ادخال في النظام المحاسبي، ولا ننسى أن هذه المخرجات هي مشتقة من ملف أو أكثر.

٨ - ٢ - ٦ ملفات الصورة الحقيقية : DUMP FILES

من أجل المحافظة على سلامة التشغيل والبيانات التي يقوم الحاسب بمعالجتها يجري عادة أخذ نسخة عن محتويات ذاكرة الحاسب بين الحين والآخر في فترة الشغل، وحفظها على صورتها الحقيقية. PHYSICAL في ملف من أجل الرجوع اليه عند الحاجة.

٨ - ٢ - ٧ ملفات مكتبة الحاسب : LIBRARY FILES

تستخدم هذه الملفات لحفظ البرامج التطبيقية أو البرامج الجزئية MODULES PROGRAMS. أو برامج الخدمة UTILITY PROGRAMS. أو بعض برامج التشغيل SOFTWARE الذي يحتاجه الحاسب.

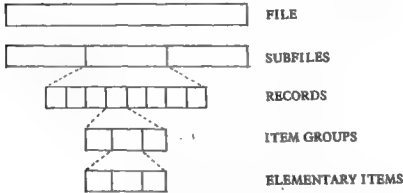
٨ - ٢ - ٨ ملفات تعديل - ثانوية : TRANSACTION FILES

تحتوي هذه الملفات على بيانات خاصة تعدل الملف الأساسي وتكون مفروزة تصاعدياً أو تنازلياً على مفتاح يطابق في توصيفه مفتاح الملف الأساسي، وكذلك تحتوي أحياناً على حقل يسمى دليل العملية TRANSACTION CODE تتم وفقه مختلف أنواع العمليات على الملف الأساسي وهذه العمليات هي : حذف سجل تعديل محتوى سجل وإضافة سجل جديد.

(٥) LIBRARY : هي عبارة عن مساحة على القرص المغنط مخصصة لاحتواء البرامج ذات الاستخدام العام لكافة المستخدمين من الحاسب، وتحتوي على عدة برامج جاهزة من الشركة الصانعة وبرامج مكتوبة من قبل إدارة تشغيل الحاسب وتكون ذات طبيعة مشتركة.

٨-٣ بنية الملف FILE STRUCTURE

- يمكن تقسيم الملف الى عدة مستويات هي :
- يمكن أن يقسم الملف الى ملفات جزئية - عند استخدام رمز معين . CODED RECORDS
- كل ملف يقسم الى سجلات RECORDS .
- السجل يقسم الى حقول FIELDS وهو أصغر وحدة يمكن معالجتها ببرنامج الكوبول .
- الحقل يقسم الى أحرف CHARACTERS (رقمي أو أبجدي) .



وفي الصفحات التالية شرح مكونات الملف (الملفات) *

٨-٣-١ الملفات الجزئية SUBFILES :

يمكن تقسيم بعض الملفات من الناحية المنطقية الى ملفات أصغر منها، فلو أخذنا على سبيل المثال ملف الحركة في الحاسبة، فهو يحتوي على مبالغ منه ومبالغ له يرمز لها برمز معين فلو فرضنا الرمز ١ يعني أن المبلغ مدین DEBIT وإذا كان الرمز ٢ يعني دائن

CREDIT، فنحن كأننا أمام ملفين أحدهما دائن والآخر مدین لكن سجلاتهما مدموجین فی ملف واحد.

ومهما یکن من الأمر فلیس المقصود من الملف الجزئی أنه یفتح بتعلیمات لوحده و یعالج و یفلق بتعلیمات مستقلة اما التقسیم من الناحیه المنطقیه LOGICAL فقط.

٨-٣-٢ السجلات : RECORDS

وهی السی تؤلف فی مجموعها الملف، وتتكون من مجموعة من الحقول FIELDS ویمكن لأكثر من حقل أن یجتمع لیكون مجموعة ضمن السجل تسمى GROUP و یشار الی توصیف السجل فی لغة كویول برقم مستوى هو 1* وللحقول المؤلفة له برقم مستوى محصور بین 2* و 49. ویمكن أن یشكل حقلاً واحداً أو مجموعة حقول FIELDS GROUP أو GROUP ITEMS ما یسمى بالمفتاح KEY لهذا السجل. والذي یستخدم لیعرّف السجل عن غیره من السجلات الموجودة ضمن الملف، أو لیظهر موقع السجل ضمن الملف كما فی الملفات النسبیه التي ستمر معنا بعد قليل. إن التعامل مع هذا الحقل الذي یسمى بالمفتاح KEY یسهل جداً التعرف علی السجل ومعالجته كما سنرى فیما بعد. فمثلاً یمكن أن یكون رقم الموظف الموجود فی السجل التابع للملف الأساسی للرواتب هو المفتاح كما یمكن لأن یكون رقم الحساب هو المفتاح فی النظام المحاسبی.

طول السجل :

یمكن تخزين السجلات ضمن الملف إما ضمن طول ثابت للسجل و یدعی عندئذ FIXED RECORD وغالباً ما یرمز له بـ F أو أن تخزن البیانات ضمن الملف فی سجلات ذات أطوال متغیره VARIABLE LENGTH و یرمز لها غالباً بـ V.

٢- السجلات ذات الطول الثابت F :

يكون شكل وطول هذه السجلات ثابت ومتساوي ضمن الملف الواحد وموقع كل حقل من حقول السجلات ثابت بالنسبة لبداية كل سجل ، والحجم الثابت للحقل يكون محسوباً على أساس أطول احتمال لورود هذا البيان ضمن الحقل ، مثلاً إذا حددنا حجم حقل الاسم في سجل ملف الرواتب بـ ٣٠ حرفاً ، فهذا يعني أنه جرى دراسة عملية لمختلف احتمالات الاسماء وتم الاتفاق بين مصمم النظم والجهة المصنعة على أكبر اسم لا يرد أكثر من ٣٠ حرفاً ، وهكذا بالنسبة لباقي الحقول .

إن هذا النوع من السجلات هو أسهل في التصميم وأيسر في المعالجة لذا نجده الأكثر انتشاراً في الاستعمال ، ولكن من الضروري الإشارة الى أنه يضيّع حجم أكبر من الفراغ ضمن وسط التخزين فليس من الضروري عملياً أن ترد كل الأسماء بطول ٣٠ حرفاً .

٣- السجلات ذات الطول المتغير V :

يحتوي على مؤشر في أول كل سجل عادة ويكون بطول 2 BYTES و يسمى VARIABLE LENGTH INDICATOR و يرمز له بـ VLI ، وفيه لا يشترط أن تكون كل السجلات متساوية الأطوال ضمن الملف .

إن استخدام عدة سجلات مختلفة الأطوال ضمن البرنامج الواحد يمكن أن يجعل البرنامج معقداً . لذلك تستخدم السجلات ذات الطول الثابت بصورة أكثر انتشاراً عن السجلات متغير الأطوال .

وتجدر الإشارة الى أنه يجب تجنب البنية المعقدة للملف ، مما يجعل المعالجة معقدة أيضاً على المبرمج وتحتاج منه لعمل تتبع وتدقيق DEBUGGING ثم تعديل لبرنامجهم مما يزيد الأمر صعوبة .

وأخيراً تجدر الإشارة الى أنه يمكن لمجموعة من السجلات - سجل أو أكثر - أن تخزن لتشكيل كتلة تسمى BLOCK لتسهيل عملية نقل البيانات من وسط التخزين الى الذاكرة أو بالعكس أثناء المعالجة ، وكذلك تزيد من فعالية أداء البرنامج بالسرعة المتزايدة مع زيادة حجم الكتلة .

وتحديد حجم الكتلة التالي يخضع لاعتبارات خاصة بنظام التشغيل مع الحاسب وطريقة تقسيم الأقراص لاحتواء الملفات .

٨ - ٣ - ٣ الحقول : FIELDS

تقسم السجلات كما مر معنا سابقاً الى حقول ذات حجم معين - ثابت أو متغير حسب نوع السجل - وتحتوي هذه الحقول على أحرف CHARACTERS قد تكون ذات طبيعة رقمية و يشار اليها بالكوبول بالرمز ٩ . وقد تكون ابجدية و يشار اليها بالرمز A ، وقد تكون مختلطة و يشار اليها بالرمز X .

٨ - ٤ تنظيم الملفات FILE ORGANIZATION

ان التنظيم الفيزيائي PHYSICAL ORGANIZATION للبيانات DATA على أوساط التخزين المختلفة (مثل الشريط أو القرص الممغنط ...) يمكن أن يعين بإيجاد علاقة بين مفتاح السجل RECORD KEY وموقع هذا السجل ضمن وسط التخزين الموجودة عليه تلك السجلات .

وهناك طرق مختلفة لترتيب السجلات داخل الملف ومن أجل كل طريقة معالجة متبعة . واختيار طريقة التنظيم تنبع غالباً الجمع بين متطلبات التحديث الناجمة (UP - TO - DATE) و بين سرعة الاستدعاء لسجلات هذا الملف .

٨ - ٤ - ١ تنظيم الملفات على أوساط تخزين متتابعة

* SERIAL STORAGE MEDIUM

من المفيد أولاً أن نبين الفرق بين عبارتي متتابع SERIAL ومتسلسل SEQUENTIAL.

إن الملف المتتابع ذلك الملف الذي فيه كل سجل يحتل الموضع التالي الشاغر ضمن وسط التخزين ولا يتبع أبداً أي تسلسل لأي من حقول السجل التي تعتبر مفاتيح لهذا السجل، ولا يوجد هنالك أيضاً أي علاقة بين قيمة الحقل ضمن الملف وبين موقعه الحقيقي PHYSICAL POSITION على وسط التخزين. وفي الملفات المتتابعة تعالج السجلات حسب تسلسلها الموجود أصلاً على وسط التخزين.

إن هذا التنظيم المتتابع يغير التنظيم المتسلسل SEQUENTIAL والذي نجد فيه أن السجلات تنظم وتعالج حسب ترتيب لفتح ضمن السجل مصمم سابقاً.

إن كلا العبارتين «المتتابع» و «المتسلسل» تصبحان مترادفتين عند استخدام الشريط المغنط MAGNETIC TAPE كوسط تخزين، فلأنه وسط تخزين متتابع فبإمكانه حفظ السجلات متتابعة ومتسلسلة. إن الشريط الورقي والبطاقات المثقبة - الشريط المغنط هي أوساط تخزين متتابعة، بينما يمكن اعتبار القرص المغنط والاسطوانة المغنطة MAGNETIC DRUM أوساط تخزين مباشرة DIRECT ACCESS MEDIUM وتسمى عادة DASD وهي اختصار لـ

DIRECT ACCESS STORAGE DEVICES

وبصورة أساسية يمكن تنظيم البيانات على أوساط التخزين المتتابعة باحدى الطريقتين :

- عشوائية : وفيها لا تكون السجلات ضمن وسط التخزين مرتبة وفق أي نظام أو مفتاح - مفهوم المتتابع SERIAL -.

- متسلسلة SEQUENTIALY : وفيها تكون السجلات مرتبة وفق تسلسل محدد

معيناً وفق مفتاح معين .

وكمثال على ذلك . إن ملف التعديل TRANSACTION يحوي على سجلات ترتيبها عشوائي SERIALY على وسط تخزين هو الشريط المغنط ، و بعد فرز سجلات هذا الملف - SORT - (أي ترتيبها وفق تسلسل مفتاح معين) يصبح هذا الملف ذا تنظيم تسلسلي .

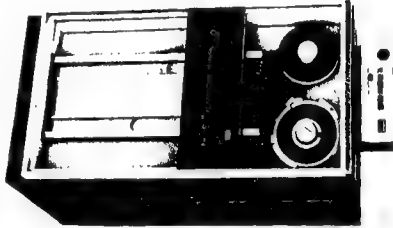
إن التنظيم العشوائي للملفات المتتابعة يعتبر أبسط نموذج من أنواع تنظيم الملفات ، وذلك بأن السجل الجديد المضاف للملف يتوضع ببساطة في آخر الملف ، بينما الملفات المتسلسلة نجد أن سجلاتها تتبع تسلسلاً معيناً حسب قيمة تصاعديّة (أو تنازليّة) لحقل معين ضمن السجل يعرف بالمفتاح KEY .

وما يجب ملاحظته أن الملفات ذات البنية المتسلسلة ORGANIZED FILES SEQUENTIALY يمكن أن تعالج بطرق غير الطريقة المتتابعة ويمكن لهذه الملفات المتسلسلة أن تخزن متتابعة على اوساط تخزين غير متتابعة مثل القرص المغنط .

إن الملفات الموجودة على وسط تخزين تتابعي SERIAL مثل تلك الموجودة على الشريط المغنط تعرف بوجود منطقة على الشريط تسمى LABEL عنوان ، وهي عبارة عن معلومات مخزنة عادة في أول وآخر ملف البيانات . وكل شريط - كوحدة فيزيائية - REEL - بكرة - يحوي على عنوان لها ، ويمكن للشريط الواحد أن يحتوي على أكثر من ملف MULTIPLE FILE VOLUME ، أو أن ملفاً واحداً يسجل على أكثر من شريط MULIVOLUME FILE وكل شريط VOLUME يحتوي على نقطة البدء أو التحميل LOAD POINT وعلامة نهاية الشريط في النهاية END OF TAPE MARKER وهي عبارة عن قطعة شريطية من الألمنيوم موضوعة في كلا طرفي الشريط ، ونقطة التحميل تشير الى بداية البيانات المسجلة على الشريط ، وعلامة نهاية الشريط تدل على مكان انتهاء البيانات المسجلة على هذا الشريط .

وصلة قارية الاشرطة

IBM TAPE DRIVE (Courtesy IBM)



| | | | | | | | | | | | | | | |
|--------------|------------|---|-------------------|---------------------|--------------------|--------------|------------------|-------------|-------------|----|--------|----|----|----|
| Faj | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Label | Faj Number | | Faj Serial Number | Faj Sequence Number | Faj Counter Number | Counter Date | Expenditure Date | Black Cover | System Code | | Amount | | | |
| Field Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Unit | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Separator | | | | | | | | | | | | | | |

شماره فای

شماره فای

شماره فای

شماره فای

شماره فای

شماره فای

شماره فای

شماره فای

شماره فای

شماره فای

شماره فای

شماره فای

شماره فای

شماره فای

شماره فای

شكل شريط

(*) COBOL A PRAGMATIC APPROACH P 279

إن وجود عبارة LABEL RECORDS ARE STANDARD في الكوبول في قسم DATA DIVISION هي من أجل الملف الموجود على وسط تخزين مغناطيسي مثل الشريط في حالتنا هذه، ونجد أن الحاسب (بوساطة OS نظام التشغيل) يعمل فحص CHECK للمعنون الداخلي للشريط INTERNAL LABEL ليتأكد من أن هل الشريط قد جهز على وحدة حاملة الأشرطة أم لا ؟.

٨ - ٤ - ٢ تنظيم الملفات على أوساط تخزين مباشرة - المعالجة *

إن أوساط التخزين المباشرة - المعالجة DASD تقدم حلاً وسطاً يجمع بين وسائل التخزين الرئيسية غير المحدودة وبين أوساط التخزين المتتالية، بتقديمها التسهيلات التالية :

- تخزين كمية كبيرة من البيانات على صلة مباشرة ON - LINE بالحاسب .
- تعالج وبتة أي جزء معطى من بيانات الملف .
- تداول هذه البيانات يتم بسرعة فائقة .

إن أوساط التخزين مباشرة - المعالجة تشمل القرص المغنط Magnetic Disk والاسطوانة المغنطة Magnetic Drum البطاقة المغنطة Magnetic Card والأقراص المغنطة المرنة Floppy Disk ، وتلك هذه الأوساط مميزات لا تملكها أوساط التخزين المتتابة، ونأخذ على سبيل المثال القرص المغنط فنجد المميزات التالية :

- إن رؤوس القراءة والكتابة يمكنها أن تمش كل الوسط في وقت قصير جداً .
- إمكانية قراءة وكتابة كتل من البيانات * * BLOCK OF DATA بحيث تعطى كل كتلة عنواناً ADDRESS خاصاً على وسط التخزين .

(*) N.C.C. P. 192

(**) BLOCK : عبارة عن مجموعة من السجلات RECORDS .

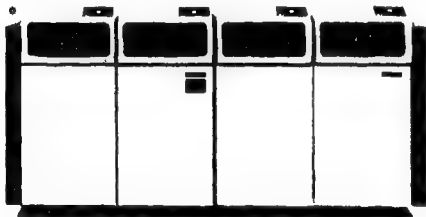
- إن السرعة الثابتة للدوران تسمح لاستغلال هذا الوسط للكتابة عليه مرات عديدة وبدقة متناهية.

وبما أن القرص المغنط من أشهر أوساط التخزين المشوائية أو مباشرة- المعالجة لذا سندرس البنية الفيزيائية للقرص المغنط بصورة مختصرة.

٨-٤-٢-١ البنية الفيزيائية للقرص المغنط

♦MAGNETIC DISK PHYSICAL CHARACTERISTIC

إن القرص المغنط DISK PACK عبارة عن وسط مغناطيسي تسجل عليه المعلومات (يمكن في بعض الأنواع أن يكون القرص مؤلفاً من جزئين جزء ثابت يدعى FIXED وجزء متحرك أو متبدل يدعى REMOVABLE) ويتألف القرص من صفائح دائرية متوازية مثبتة بمحور مركزي و يركب هذا القرص على وحدة حاملة الأقراص المغنطة تسمى DISK DRIVE والتي هي عبارة عن الوحدة التي تقرأ وتكتب من وعلى القرص المغنط و يظهر الشكل التالي القرص المغنط وكذلك وحدة حاملة الأقراص المغنطة :



وحدة حاملة الأقراص المغنطة من نوع I.B. M

(*) COBOL A PRAGMATIC APPROACH GRAUER and CRAWFORD 1978 P 269



IBM 2314 Disk Pack (Courtesy IBM)

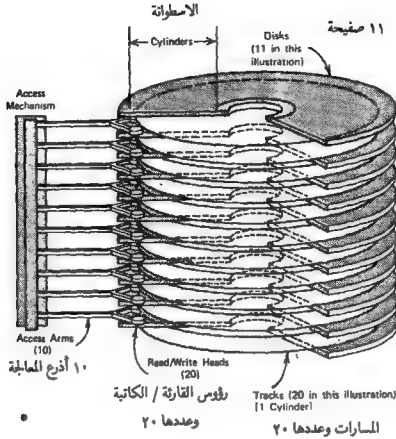
قرص مخفض من نوع I.B.M

2314

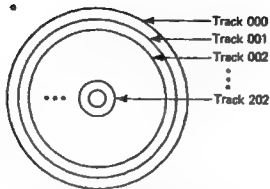
نلاحظ من شكل القرص المخفض أنه مؤلف من ١١ صفيحة ومغطى بغطاء واقى ويمكن التسجيل على كل صفيحة أي السطح العلوي والسطح السفلي ما عدا الوجه العلوي للصفيحة الأولى - العلوية - والسطح السفلي للصفيحة - السفلية - أي يمكن استخدام عشرين سطح للتسجيل .

وتتم عملية الكتابة والقراءة بواسطة رؤوس قارئة - كاتبة تدخل بين الصفائح كما تدخل أسنان المشط بين الشعر عند تصفيفه ، وهذه الرؤوس مثبتة على ذراع كما يلاحظ من الشكل التالي :

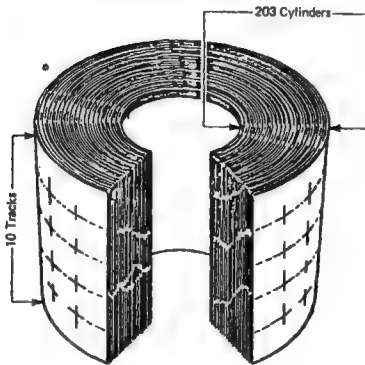
(هـ) هذا القرص من نوع IBM 2314 ويمكن أن توجد أقراص مختلفة الأحجام بعدد سطوح أو صفائح أكثر أو أقل وليس هذا الرقم ثابتاً .



إن كل سطح تسجيل مقسم الى دوائر متحدة المركز كل دائرة تسمى مساراً TRACK والتي تسجل عليها المعلومات ولكل مسار له نفس قدرة التخزين لباقي المسارات وهكذا فإن المسار الداخلي القريب من المركز له نفس قدرة التخزين للمسار البعيد عن المركز.



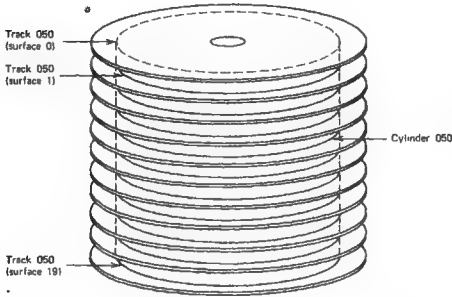
شكل مواجه لصفحة تظهر عليها المسارات



شكل آخر يوضح المسار والاسطوانة Cylinder

← STERN (•)

إن مجموع مسارات أفقية معينة في القرص كله تسمى بـ CYLINDER مثال : المسار رقم ٥٠ في السطح رقم ١ والمسار رقم ٥٠ في السطح رقم ٢ والمسار رقم ٥٠ في السطح رقم ٣ ... والمسار رقم ٥٠ في السطح رقم ٩ تؤلف في مجموعها (عمودياً) مفهوم الـ CYLINDER أنظر الشكل التالي :



ويمكن أن نذكر أنه في بعض الحاسبات يقسم كل مسار TRACK الى أجزاء أصغر منه تسمى قطاعاً SECTOR ، وهذا القطاع هو الذي تسجل عليه البيانات وهو يعادل أو يقارب مفهوم كتل البيانات BLOCKS لكن هنا التقسيم من الناحية الفيزيائية .

٨ - ٢ - ٢ القدرة التخزينية للأقراص المغنطة CAPACITY**

إن الحد الأقصى لاستيعاب وحدة الاقراص المغنطة يحدد بثلاثة عوامل :

أ - كثافة التسجيل (عدد الأحرف BYTES التي تسجل على المسار TRACK) .

(*) STERN

(**) COBOL A PRAGMATIC APPROACH 1978 Grauer and Crowford P 297 - 298

- ب - عدد أسطح التسجيل RECORDING SURFACE (عدد المسارات بكل CYLINDER).
- ج - عدد المسارات على أسطح التسجيل (عدد الـ CYLINDERS).
- وتزداد قدرة التخزين بزيادة أي عامل من العوامل الثلاثة السابقة.

٨ - ٤ - ٢ - ٣ مفهوم المساحة الفائضة OVERFLOW AREA

إن المنحنى الطبيعي للملفات الأساسية MASTER FILES هي التوسع في الحجم (expand). ويمكن أن يتم التوسع أيضاً بزيادة عدد الملفات المخزنة على القرص. أو بزيادة حجم هذه الملفات، وفي كلا الحالتين يمكن أن يصل القرص الى حد الاشباع أو حالة الفيضان OVERFLOW، لذا نجد أنه في كل CYLINDER أو كل TRACK تخصص مساحة تسمى OVERFLOW AREA يمكن أن تساعد على حل مؤقت لمشكلة التوسع، فالملفات عند انشائها CREATION يُحدد لها مساحة معينة، فإذا وصل هذا الملف لحده الاشباع أي استهلكت المساحة المخصصة له (أو في بعض حالات التنظيم العشوائي أن أكثر من مفتاح يكون له نفس عنوان التخزين) لذا يتم تسجيل السجل الجديد في المساحة الإضافية OVERFLOW AREA مع تعريف الملف بعنوانه في تلك المساحة.

وفي أحيان كثيرة يكون «الفيضان» للمعلومات بهكان معين في القرص وباقي الأماكن تكون شاغرة. هذه الحالة تنشأ من أنه يمكن لبعض الملفات أن تكون نشطة - أي يجري عليها عمليات TRANSACTIONS بصورة دورية فتمتلئ المساحة المخصصة للملف وكذلك المساحة الإضافية في حين تكون بعض الملفات خاملة أي أن العمليات التي تجري عليها قليلة، أو أنها تكون ذات صفة مصدرية قليلة التعديل، لذا فتكون مساحتها الإضافية فارغة، وجزء من المساحة الأصلية أيضاً.

لذا نجد أن أحد الحلول المطروحة للاستغلال الأمثل للفرغ SPACE الموجود على القرص هي إعادة تنظيم القرص REORGANIZATION . أو على الأقل إعادة تنظيم بعض الملفات الموجودة ضمنه بإعطاء الملفات النشطة حجماً أكبر والملفات الخاملة حجماً أقل .

وتجدر الإشارة أنه عند إعادة تنظيم القرص يستعان بوسط تخزيني آخر وسيط قد يكون قرصاً آخر أو شريطاً مغناطيسياً .

الآن وبعد أن عرفنا البنية الفيزيائية للقرص سندرس بشيء من التفصيل تنظيم الملفات على هذا الوسط .

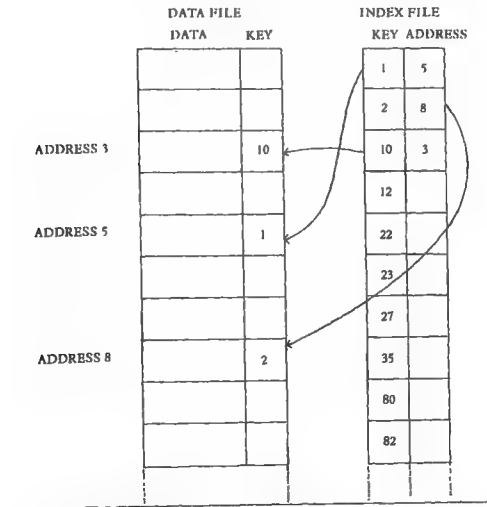
يمكن أن تنظم الملفات على وسط التنظيم مباشرة - المعالجة بأحد أربعة طرق :

- | | |
|----------------------------------|------------------|
| STANDARD SEQUENTIAL ORGANIZATION | أ - تنظيم متسلسل |
| INDEXED ORGANIZATION | ب - تنظيم مفهرس |
| DIRECT ORGANIZATION | ج - تنظيم مباشر |
| RELATIVE ORGANIZATION | د - تنظيم نسبي |

أ - التنظيم المتسلسل : هو مشابه تماماً للتنظيم الموجود على أوساط تخزينية متتابعة مثل الشريط ، فكل سجل يأتي مباشرة بعد السجل الذي يسبقه إما بترتيب معين حسب مفتاح - متسلسل - أو بشكل عشوائي - متتابع - ويتم معالجته تماماً كما لو أنه على شريط ممغنت ، وله عنوان بداية ونهاية أيضاً - LABEL - . إن هذا التنظيم يستغل المساحة المعطاة للسلف استغلالاً أمثل ولا يترك فراغات بين السجلات والتحديث عليها يتم مثل التحديث على نفس الملفات كما لو كانت على شريط ممغنت .

ب- تنظيم الملفات المتسلسلة المفهرسة *Indexed sequential files organization

إن الملفات المتسلسلة المفهرسة هي الطريقة في التنظيم التي تسمح للمبرمج معالجة هذا النوع من الملفات والموجود على القرص المغنط بشكل عشوائي RANDOM. وتعتمد هذه الطريقة في التنظيم على إيجاد ملف جزئي يلحق بالملف الأصلي DATA FILE- أي ملف البيانات، هذا الملف الجزئي يحتوي على مفاتيح سجلات الملف RECORD KEYS مرتبة بشكل متسلسل.



(هـ) تعرف أحياناً في بعض الحاسبات بالرمز ISAM وفي I.B.M تحت نظام تشغيل OS/MVS تدعى VSAM.

ويحتوي بالإضافة الى ذلك عنوان سجل كل مفتاح على القرص .

ان كلاً من الملف الجزئي - الفهرس - وملف البيانات يؤلفان ملفاً واحداً يسمى Indexed sequential file وما هذا التقسيم إلا ضمني ، أما ما يخص مستخدم الحاسب USER أو المبرمج Programmer في التوصيف وإنشاء الملفات وغيرها فهو يتعامل مع اسم واحد فقط وهو اسم الملف المفهرس في البرنامج .

ولتوضح لنا الصورة بشكل أكبر نضرب المثل التالي :
إذا أردنا البحث عن موضوع معين ضمن كتاب ؛ فإما أن نقرأ الكتاب صفحة صفحة حتى نصل للبحث المطلوب . أو ننظر في فهرس الكتاب Index ونأخذ منه رقم الصفحة - العنوان - ونفتح مباشرة على البحث المطلوب .
ولا ننسى أن كلاً من المعلومات في الكتاب والفهرس هما تحت عنوان واحد .

أخيراً نشير الى أنه لكل سجل لا بد أن يكون هناك مفتاح ولهذا المفتاح قيمة وحيدة ضمن الملف UNIQUE و يسمى PRIME KEY المفتاح الأولي أو الأسامي ؛ ويمكن للسجل أن يعرف بمفاتيح ثانوية وثالثية حسب امكانيات كل حاسب ويمكن لهذه المفاتيح - الثانوية ALTERNATE KEYS أن تكون قيمتها مكررة DUPLICATES في الملف .

ج تنظيم الملفات المباشرة *DIRECT FILE ORGANIZATION

يعتمد هذا الأسلوب من التنظيم على إجراء بعض العمليات الحسابية RANDOMIZING على الحقل الذي يسمى مفتاح سجل RECORD KEY ومن ناتج

(*) هذا النوع من التنظيم غير شائع في كل الحاسبات ويستعاض عنه بالتنظيم المفهرس التسلسل INDEXED SEQUENTIAL ، و يستخدم هذا النوع من التنظيم في أنظمة قواعد البيانات DATA BASE بصورة أساسية .

هذه العملية الحاسوبية يعرف عنوان السجل على القرص الممغنط حسب رقم الاسطوانة CYLINDER والمسار TRACK ويخزن في ذلك العنوان. إن الطريقة هذه لا تعتمد على قراءة متسلسلة من ملف البيانات أو من ملف المفهرس وإنما على هذه العملية الحاسوبية لذلك سميت DIRECT لأنه مباشرة ينفذ للعنوان المطلوب.

وتجدر الإشارة أن هذه العمليات الحاسوبية RANDOMIZING ليست بالصيغة البسيطة التي نتصورها، إنما تعتمد أساساً على إيجاد صيغة رياضية معقدة من أجل إيجاد عنوان وحيد UNIQUE لكل سجل جديد مضاف الى الملف. وكثيراً من الأحيان نجد أن الوقت اللازم لانجاز هذه العملية أكبر من الوقت المصروف على البحث في ملف مفهرس، لذلك نجد في أغلب التطبيقات أن الملفات المفهرسة Indexed sequential files أكثر استعمالاً من Direct files.

د- تنظيم الملفات النسبية RELATIVE FILE ORGANIZATION

في هذا التنظيم نجد أن كل سجل معرف برقم ذي علاقة نسبية مع بداية الملف يسمى RELATIVE RECORD وهو وحيد ضمن الملف. إن هذا التنظيم مسموح به فقط على القرص الممغنط. إن رقم السجل رقم صحيح INTEGER وأكبر من الصفر والسجلات تخزن وتستدعى اعتماداً على هذا الرقم وتبدأ من الرقم ١. فالسجل رقم ١٠ هو ذاك السجل ذو الترتيب العاشر في الملف؛ والسجل الذي رقمه ٢١٥ هو ذاك السجل الذي ترتيبه مئتان وخمسة عشر في الملف والسجل رقم ٢ هو السجل ذو الترتيب الثاني في الملف ... وهكذا.

٨- ٥- أساليب معالجة الملفات FILE ACCESS METHODS

تختلف أساليب معالجة الملفات باختلاف أوساط التخزين، فكما مر معنا في الجزء السابق (٤) نجد أن هنالك نوعين من أوساط التخزين :

أ- أوساط تخزين متتابعة SERIAL STORAGE DEVICES

ب - أوساط تخزين مباشرة المعالجة DIRECT ACCESS STORAGE DEVICES

٨ - ٥ - ١ معالجة الملفات المخزنة على أوساط تخزين متتابعة

ان الملفات المتسلسلة التنظيم SEQUENTIAL ORGANIZATION . المخزنة على أوساط تخزين متتابعة مثل الشريط المغنط أو البطاقات المثقبة تعالج بصورة بسيطة جداً، إذ تتم قراءة كل سجل وتجري عليه العمليات المطلوبة . نفس الشيء ينطبق على الملفات المتتابعة Serial files .

ومن أجل توصيفها ومعالجتها يرجع للفصول السابقة من أجل شرح هذه التعليمات أو التوصيف .

ونذكر بصورة مختصرة وسريعة التعليمات الأساسية التي تعالج هذه الملفات :

١ - تعليمية فتح الملف

| | | |
|-------------|--|-------------------|
| <u>OPEN</u> | $\left\{ \begin{array}{l} \text{INPUT} \\ \text{OUTPUT} \\ \text{EXTEND} \end{array} \right\}$ | file name 1, ..., |
|-------------|--|-------------------|

٢ - تعليمية القراءة

READ file - name [INTO identifier] AT END

imperative statements

٣ - تعليمية الكتابة

WRITE Record - name [FROM identifier]

[AFTER ADVANCING n LINES].

٤ - تعليمية إغلاق الملف

CLOSE file name 1,

٥ - ونذكر توصيف الملف في

INPUT - OUTPUT SECTION.

(٥) الفصل الثاني من القسم الأول

FILE - CONTROL.

SELECT file - name

ASSIGN TO implementor - name

نوع تنظيم الملف تسلسلي
اسلوب المعالجة تسلسلي

ORGANIZATION IS SEQUENTIAL,

ACCESS MODE IS SEQUENTIAL.

فعند طباعة ملف مثلاً، يقرأ الملف وتحرك الحقول منه الى حقول الطباعة ويتم الكتابة على الطابعة باستخدام تعليمات :

OPEN

READ

MOVE

WRITE ...

وهذا أبسط شكل معالجة ؛ لكن عندما نريد الاستفسار عن سجل محدد عندها يجب أن نقرأ الملف سجلاً سجلاً حتى نحصل على السجل المطلوب فلو كان عندنا ١٠٠٠ الف سجل في ملف الرواتب وأردنا الاستفسار، أو طباعة الموظف رقم ٩٠٠ الموجود سجله في الملف ؛ فيجب علينا أن نقرأ ٨٩٩ سجلاً قبله حتى نصل اليه ثم نطبع السجل أو نعالجه حسب المطلوب ؛ إن المشكلة تزداد صعوبة إذا طلب منا موظف آخر ذو رقم ٢٠٠ فالتعليمات المتوفرة واسلوب تنظيم الملف ومعالجته لا تسمح بالقراءة للخلف أي كل قراءة تأخذ السجل الذي يلي السجل الحالي، فالحل الوحيد لمعالجة الموظف رقم ٢٠٠ بعد أن عالجنا الموظف رقم ٩٠٠ هو إغلاق هذا الملف بتعليمية CLOSE ثم إعادة فتحه بتعليمية OPEN ثم القراءة ١٩٩ مرة حتى نصل للسجل رقم ٢٠٠ المطلوب (يتم إغلاق وفتح الملف حتى نحصل على السجل الأول من الملف).

٨ - ٥ - ١ - ٢ تحديث الملفات المتسلسلة UPDATE SEQUENTIAL FILES

يقصد بعملية التحديث هو تعديل محتوى الملف بما يتناسب مع آخر معلومات صحيحة عن الملف .

فلو أخذنا ملف البيانات الشخصية في نظام الشئون الشخصية PERSONNEL نجد أنه يحتوي على سجلات الموظفين ، كل سجل يحتوي على حقول تخص حالة الموظف ، اسمه ، رقمه ، عنوانه ، حالته الاجتماعية ، ... الى ما هنالك من معلومات .

إن الملف بحد ذاته عرضة للتغير كأن يعين موظف جديد ، إذا سنضيف سجلاً الى الملف ؛ أو قد يترك العمل موظفاً آخر ، أي سنحذف سجلاً من الملف ، وقد تكون المعلومات نفسها بحاجة لتغيير ، لو فرضنا أن موظفاً غير عنوانه ، أو تزوج أو طلق ، فالحقول الموجودة ضمن الملف تحتاج الى تحديث .

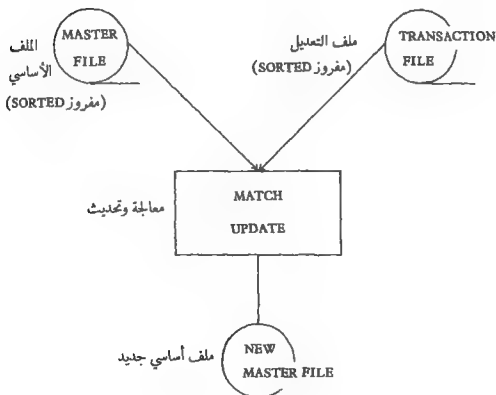
إن الامكانيات المتاحة في الملفات المتسلسلة والموجودة على أوساط تخزين متتابعة مثل الشريط المغنط لا تسمح بتبديل محتوى الملف ، أو حذف سجل مباشرة منه أو اضافة سجل مباشرة اليه .

إذاً يجب علينا الاستعانة بملفين آخرين ، ملف فيه الوقوعات أو التبديلات TRANSACTIONS ويسمى ملف التعديل و يوجد فيه رقم السجل - الموظف - المراد معالجته ، ونوع العملية المرغوب فيها وتسمى غالباً TRANSACTION CODE وهي حقل ضمن سجل ملف التعديل ؛ و ملف آخر هو ملف وليد عملية دمج الملفين السابقين ملف الشئون الشخصية مع ملف التعديل ، وهذا الملف الوليد يحتوي على السجلات التي لم يطرأ عليها أي تعديل الموجودة في ملف الشئون الشخصية بالاضافة الى التعديلات المرغوب فيها والقادمة من ملف التعديلات مثل اضافة سجلات جديدة ، أو سجل تعديل لسجل أساسي .

إن الملف الوليد هذا هو عبارة عن ملف أساسي جديد NEW MASTER FILE لذا يجري إلغاء الملف الأساسي القديم - أو الاحتفاظ به بعيداً عن المعالجة الحالية - وتعاد تسمية الملف الوليد بالملف الأساسي ويصبح هذا الملف هو الملف الأساسي MASTER FILE ويكون محدثاً UP - TO - DATE .

وتجدر الإشارة أنه يجب قبل البدء في معالجة الملفات السابقة - الأصلي القديم وملف التعديل - أن يكونا مفروزين SORTED على مفتاح معين في كلا الملفين في حالتنا هنا على رقم الموظف، وعلى نفس الاتجاه (تصاعدي أو تنازلي) .

أنظر البرنامج التالي والذي فيه يتم تحديث ملف أساسي من ملف حركة .



```

*
IDENTIFICATION DIVISION.
PROGRAM-ID. TXT22.
AUTHOR. KHAYAT-JNAID.
*
* يقوم هذا البرنامج بقرائة ملف اساسي وملف
* تعديل ثم ينفذ ملف اساسي جديد
*
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT MASTRF ASSIGN TO MSTF
    ORGANIZATION SEQUENTIAL;
    ACCESS MODE SEQUENTIAL.
    SELECT TRANSF ASSIGN TO TRNF
    ORGANIZATION SEQUENTIAL;
    ACCESS MODE SEQUENTIAL.
    SELECT NMASTRF ASSIGN TO NMSTF
    ORGANIZATION SEQUENTIAL;
    ACCESS MODE SEQUENTIAL.
DATA DIVISION.
FILE SECTION.
* توصيف الملف الاساسي
FD MASTRF LABEL RECORD IS STANDARD
DATA RECORD IS M-REC.
    01 M-REC.
        03 M-NO PIC X(5).
        03 M-DATA PIC X(35).
* توصيف ملف التعديل
FD TRANSF LABEL RECORD IS STANDARD
DATA RECORD IS T-REC.
    01 T-REC.
        03 T-NO PIC X(5).
        03 T-DATA PIC X(34).
* يعتبر هذا هو حقل رمز العملية
    03 T-CODE PIC X.
* توصيف الملف الاساسي الجديد
FD NMASTRF LABEL RECORD IS STANDARD
DATA RECORD IS N-REC.

```

```

01      N-REC.
      03      N-NO          PIC X(5),
      03      N-DATA       PIC X(35).

```

```

*
PROCEDURE DIVISION.
      OPEN          INPUT  MASTRF TRANSF
                   OUTPUT  NMASTRF.
*      القراءه من الملفين
*
      PERFORM READ--MSTR THRU EXIT1.
      PERFORM COMPAR THRU EXIT-LOOP
            UNTIL T-NO = M-NO
            AND    T-NO = HIGH-VALUE.
      CLOSE MASTRF
            TRANSF
            NMASTRF.
      STOP          RUN.

```

*

```

*
*      مقارنه قيمه المحتاج في السجلين
*      وكتابه السجل الذي محتاجه امحى على
*      ملف الاخراج ثم القراءه منه من جديد.
*      اما في حاله المساواه فاداك كان رمز
*      العمليه D يعنى حذف فيقرأ من
*      الملفين من جديد (اي يتجاوز الكتابه)
*      اما حاله التعديل C فينقل محتوى السجل
*      من ملف التعديل الى الاساسي الجديد.
*

```

```

COMPAR.
      IF          M-NO LESS THAN T-NO
            WRITE N-REC FROM M-REC
            PERFORM READ--MSTR
      ELSE
            IF T-NO LESS THAN M-NO AND
               T-CODE = 'A'
            PERFORM MOVE--2 THRU EXIT2
      ELSE
            IF T-NO = 'D'
            PERFORM READ--TRNS THRU EXIT1

```

```

ELSE
  IF T-NO = 'C'
    PERFORM MOVE-2 THRU EXIT1.
EXIT-LOOP.
EXIT.
*
MOVE-2.
  MOVE T-NO      TO N-NO.
  MOVE T-DATA    TO N-DATA.
  WRITE N-REC.
*   الفرقة من ملف التعديل
READ-TRNS.
  READ TRANSF AT END
  MOVE HIGH-VALUE TO T-NO.
EXIT2.
EXIT.
*   الفرقة من الملف الأساسي
READ-MSTR.
  READ MASTRF AT END
  MOVE HIGH-VALUE TO M-NO.
EXIT1.
EXIT.
*
```

إن الفكرة الأساسية في هذا البرنامج، برنامج التحديث هي القراءة من الملف الأساسي ومن ملف التعديل ومقارنة المفتاحين وتكون أمام الاحتمالات التالية :

- أ - فإذا كان مفتاح الملف الأساسي أصغر من مفتاح ملف التعديل فهذا يعني أن هذا السجل الأساسي ليس عليه أي تعديل فينقل إلى الملف الجديد - الوليد - ويكتب عليه ويرجع للقراءة من الملف الأساسي ثانية .
- ب - إذا كان مفتاح ملف التعديل أصغر من مفتاح الملف الأساسي فهذا يعني أن ملف التعديل يحمل سجلاً لا يوجد مثيل له في الملف الأساسي أي سجلاً جديداً فينقل هذا السجل من ملف التعديل إلى الملف الجديد الوليد ويكتب ويقرأ من ملف التعديل ثانية ، على شرط أن يكون رمز العملية 'A' .

جـ - في حال كون المفتاحين متساويين فهذا يعني أن للسجل الأساسي سجل تعديل وهذا التعديل يكون أحد أمرين :

جـ - ١ أما أن رمز العملية TRANSACTION CODE في سجل التعديل يحتوي على رمز الحذف (و بالفرض) الحرف D عندها نعاود قراءة الملفين من جديد ونكون بذلك قد مررنا على السجل الموجود في الملف الأساسي دون نقله للملف الوليد - BY PASS - أي حذفناه .

جـ - ٢ أو أن رمز العملية TRANSACTION CODE في سجل التعديل يحتوي على رمز التغير C مثلاً، عندها يكون سجل ملف التعديل يحتوي على معلومات جديدة عن نفس السجل الموجود في الملف الأساسي، لذلك ننقل المعلومات من سجل التعديل الى الملف الوليد - الجديد - ونعاود قراءة الملفين معاً .

قد تصادف بعض المبرمجين أو الطلاب مشكلة نهاية الملفات END OF FILES فلو انتهى ملف التعديل أولاً فيجب نقل المتبقي من الملف الأساسي الى الملف الجديد وكتابته (أي سجلات ليس عليها تعديلات) أما لو انتهى الملف الأساسي أولاً فهذا يعني أن المتبقي من ملف التعديل هو سجلات جديدة للاضافة لذلك ننقل الى الملف الجديد - الوليد - وكتبت، إن معالجة ذلك في برنامج كويول قد يسبب إرباكاً عند غلق الملف المنتهي والبدء بروتين جديد لمعالجة المتبقي من أحد الملفين . لكن اذا استعملنا القيمة العليا HIGH-VALUE •• للمقارنة (يفرض أن الملفين مفروزان تصاعدياً) عند انتهاء أحد الملفين فيكون الملف الآخر غير المنتهي حالة أصغر بالنسبة للملف المنتهي

(•) قد تنص المسألة إلى أنه يوجد أكثر من سجل تعديل للسجل الأصلي عندها نعاود القراءة من ملف التعديل فقط، وهذا الأسلوب غالباً ما يكون في النظم الحاسوبية .

(••) قيمة معرفة في الحاسب ولا داعي لوصفها في WORKING-STORAGE وقائل في تعريفها كلمة SPACE أو ZERO مع اختلاف محتواها طبعا، وهي قيمة ثابتة في لغة كويول تماثل أعلى قيمة يمكن وصفها في الذاكرة وتسمى . FIGURATIVE CONSTANT

والذي يحوي على القيمة العليا HIGH - VALUE فنعود للحالة أ أو ب ولا حاجة لكتابة تعليمات جديدة - أنظر البرنامج - وعند تساوي مفتاحي الملفين مع HIGH - VALUE يكون قد انتهى الملفان فنقلهما ونهيه العمل .

يمكن أن نشير الى أنه قد يطلب عمليات مختلفة على برنامج تحديث الملف الأساسي كالكشف عن التكرار في ملف التعديل وغيرها من طلبات تعالج عندها حسب المطلوب .

إن اسلوب المعالجة المتسلسلة هو الأسلوب الوحيد المتوفر في معالجة الملفات الموجودة على أوساط تخزين متتابعة مثل الشريط المغنط والبطاقات المثقبة ، بينما نجد أن هنالك أكثر من طريقة أو أسلوب معالجة على الملفات الموجودة على أوساط تخزين مباشرة - المعالجة DASD ، كما سنرى في الفصل التالي .

وأخيراً تجدر الإشارة الى أن هذا النوع من المعالجة يسمى غط المعالجة غير المباشرة BATCH MODE أو OFF - LINE والتي من مساوئها البطء في استدعاء سجل معين وتعقيد العمليات التي يجنبها المشغل بنقل ملفات من وإلى الحاسب وعدم امكانية التعديل على نفس الملف .

٨- ٥- ٢ معالجة الملفات المخزنة على أوساط تخزين مباشرة - المعالجة DASD

تقسم أنواع المعالجة التي يمكن أن تجري على الملفات الى قسمين أساسيين :

١ - المعالجة المتسلسلة SEQUENTIAL ACCESS

٢ - المعالجة العشوائية (المباشرة) RANDOM ACCESS (DIRECT)

ويمكن أن نعالج ملفاً معيناً بالطريقتين معاً و بنفس البرنامج و يسمى عندئذ اسلوب المعالجة ب المعالجة الحركية :

٣ - المعالجة الحركية DYNAMIC ACCESS

٨-٢-٥-١١ المعالجة المتسلسلة :

وفيهما تتم معالجة سجلات الملف بصورة متسلسلة وكل سجل يقرأ ثم يعالج ، وتطبق على الملفات ذات التنظيم التسلسلي ، وذات التنظيم المهرس ، وذات التنظيم المباشر DIRECT ، وكذلك الملفات ذات التنظيم النسبي .

٨-٢-٥-١٢ معالجة الملفات المتسلسلة SEQUENTIAL FILES

إن أسلوب وتوصيف المعالجة المتسلسلة للملفات المتسلسلة هو نفس الأسلوب المتبع في المعالجة المتسلسلة للملفات الموجودة على وسط تخزين متتابع والموصوفة في الفقرة ٨-٥-١١ وأسلوب التحديث عليها هو نفسه الموصوف في الفقرة ٨-٥-١٢ ، لذلك راجع الفقرتين المذكورتين من أجل معرفة معالجة الملفات المتسلسلة تسلسلياً .

٨-٢-٥-١٣ معالجة الملفات المهرسة INDEX SEQUENTIAL FILES

إن أسلوب المعالجة للملفات المتسلسلة والمهرسة هو قريب من المعالجة المتسلسلة للملفات المتسلسلة مع بعض الاختلافات التي سندكرها هنا :

- في قسم IDENTIFICATION DIVISION لا نلاحظ أي اختلاف بالتوصيف .

- في قسم ENVIRONMENT DIVISION نلاحظ أنه في FILE-CONTROL أصبح التوصيف كالتالي :

SELECT file - name ASSIGN TO implementor - name,

ORGANIZATION IS INDEXED,

ACCESS MODE IS SEQUENTIAL

RECORD KEY IS data - name - 1 e

(e) بعض الحواشي مثل TI تشترط بتوصيف المفتاح الألى أن يكون من نوع ALPHANUMERIC .

[ALTERNATE RECORD KEY IS data - name - 2]

[WITH DUPLICATES] [FILE STATUS IS data - name - 3].

نلاحظ من التوصيف السابق :

أن نوع تنظيم الملف هو فهرس ORGANIZATION IS INDEXED واسلوب
المعالجة هو متسلسل ACCESS MODE IS SEQUENTIAL وبما أنه فهرس إذاً يجب
أن يكون له مفتاح KEY و بالتالي نضع اسم المفتاح مكان 1 - data - name - نذكر أن
المفتاح هو حقل في السجل ضمن الملف - هذا المفتاح طبعاً معرف في قسم الملفات
DATA DIVISION ضمن توصيف السجل . و 1 - data - name - تدل إذاً على
المفتاح الأساسي الأولي Prime key للملف .

ويمكن أن تكون هنالك مفاتيح ثانوية فتوصف عندئذ في عبارة :

ALTERNATE RECORD KEY IS data - name - 2,

وإذا لم يوجد للسجل سوى مفتاح أساسي واحد فلا داعي لتوصيف المفتاح
الملحق . أما عبارة WITH DUPLICATES فتفيد بتعريف الحاسب أن هذا المفتاح
الثانوي - وليس الأساسي قطعاً - ليس وحيداً بالملف بل مكرراً DUPLICATE . فإذا
لم توضع هذه العبارة اعتبر المفتاح الثانوي وحيداً أيضاً UNIQUE ، فإذا صادف أن
وصفنا المفتاح الثانوي بأنه وحيد وأردنا إضافة سجل للملف له نفس قيمة المفتاح
الثانوي حصلنا على حالة خطأ .

يمكن تعريف أكثر من مفتاح ثانوي لنفس الملف ؛ إما العدد يختلف من حاسب
لآخر فنجد في TI مثلاً بحدود ١٣ مفتاح ثانوي .

- في قسم البيانات DATA DIVISION لا يوجد أي توصيف خاص .

- في قسم الإجراءات PROCEDURE DIVISION

(هـ) بعض الحواسيب لا تتطلب توصيف المفتاح الأولي أن يكون من نوع ALPHANUMERIC .

ولنستعرض التعليمات الأساسية التي يمكن أن يحصل فيها بعض التغيير :

أ - تعليمة فتح الملف OPEN

$$\bullet \text{OPEN} \left\{ \begin{array}{l} \text{INPUT} \\ \text{OUTPUT} \\ \text{I - O} \end{array} \right\} \text{file name 1, ...}$$

نلاحظ أنه يمكن فتح الملف المفهرس إما بشكل INPUT وبالتالي نستطيع أن نقرأ منه ولا نستطيع أن نكتب عليه ؛ أو بشكل OUTPUT أي نستطيع أن نكتب عليه دون أن نقرأ منه ؛ والشكل الثالث وهو جديد I - O أي INPUT - OUTPUT أي يمكن أن نقرأ منه ونكتب عليه بنفس البرنامج وهذه ميزة الملفات المفهرسة . بعد تنفيذ هذه التعليمة بنجاح يكون المؤشر POINTER الخاص بالملف متوضعا عند بداية أول سجل في الملف .

ونذكر هنا أنه من جملة الأخطاء التي يواجهها المبرمج عند تنفيذ برنامج يحوي على سجل مفهرس أن تعليمة OPEN لا تنفذ بنجاح ويكون على الغالب توصيف الملف في عبارة SELECT أو في عبارة FILE DESCRIPTION مغايراً للمعلومات التي أعطيت للحاسب عند انشاء الملف فيزيائياً على القرص . كأن يكون طول المفتاح مغايراً للطول الموصف في البرنامج ، أو أن يكون هناك تكرار في المفتاح الثانوي ولم يذكر ذلك المبرمج في عبارة ALTERNATE KEY WITH DUPLICATES . لذلك يجب الانتباه لتوصيف الملف ومفاتيحه في البرنامج بأن تكون مطابقة تماماً للمعلومات التي أعطيت للحاسب عند انشاء هذا الملف فيزيائياً على الوسط .

ب - تعليمية القراءة READ

نجد أن الشكل للتعليمية

♦ READ file name NEXT RECORD [INTO identifier]
[AT END imperative - statement.]

هذه التعليمية تطبق عندما يراد القراءة من ملف متسلسل ومفهرس بطريقة متسلسلة
ACCESS MODE SEQUENTIAL.

نلاحظ هنا أن الفرق بين هذه التعليمية وتعليمية READ المستخدمة في المعالجة المتسلسلة للملفات المتسلسلة - غير المفهرسة - هو وجود عبارة NEXT أي السجل التالي .
هذه التعليمية تنفذ إذا كان الملف مفتوح بشكل INPUT أو O - I فقط .

ج - تعليمية الكتابة WRITE الشكل العام للتعليمية :

♦ WRITE record - name [FROM identifier]
INVALID KEY imperative - statement

تستخدم هذه التعليمية للكتابة على ملف مفهرس ذي اسلوب معالجة متسلسلة أو عشوائية أو حركية . ويشترط في حالتنا هذه اي المعالجة المتسلسلة للملفات المفهرسة بأن تكون قيم مفتاح السجل أعلى من قيمة مفتاح السجل الموجود قبله أي لو كان آخر سجل موجود في الملف قيمة مفتاحه ١٢٥ فيجب أن تكون قيمة المفتاح للسجل المضاف أعلى من ١٢٥ وإلا فتتشكل حالة INVALID KEY كما يجب أن يكون الملف مفتوحاً بشكل OUTPUT أو O - I .

مثال :

WRITE OUT - REC INVALID KEY

DISPLAY "RECORD NOT ADDED" GO TO ERRRTN.

DISPLAY "SUCCESSFUL INSERTION".

MOVE ...

من المثال نلاحظ أنه اذا تمت الاضافة بنجاح يظهر المبرمج مثلاً رسالة تشعر بذلك ويتابع المعالجة ؛ وإذا لم تتم وتشكل خطأ نجد أن المبرمج أظهر خطأ وذهب الى فقرة ببرنامجها اسمها ERR - RTN ، والذهاب لهذه الفقرة لا يكون إلا عندما تتشكل حالة الخطأ .

د. تعلية اعادة الكتابة *REWRITE

تستخدم هذه التعلية لتعديل محتوى سجل معين ضمن الملف المفهرس . ففي الملفات ذات التنظيم التسلسلي نجد أنه لا نستطيع الكتابة عليها ، أو اعادة الكتابة إنما نستخدم ملفاً ثالثاً ليكون هو حاصل التعديل - انظر الفقرة ٨ - ٥ - ١ والمثال - بينما هنا في الملفات المفهرسة نستطيع أن نضيف سجلات جديدة كما نستطيع أن نبدل محتوى سجل معين ونعيد كتابة السجل القديم بمحتواه الجديد . أما التعديل فقد يأتي من سجل آخر أو عن طريق الشاشات .

والشكل العام للتعلية :

REWRITE record - name [FROM identifier].

يشترط لتنفيذ هذه التعلية أن يكون الملف مفتوحاً بشكل 0 - 1 .

مثال :

REWRITE IN - REC.

*START تعليمية البداية

هذه التعليمية توفر للمبرمج الأساس الذي سيبدأ منه المعالجة المتسلسلة للسجلات ضمن الملف.

فلو أراد المبرمج أن يبدأ المعالجة من قيمة معينة للمفتاح وليس من أول الملف فإن استخدام تعليمية START تحدد موقع بدء المعالجة.
الشكل العام لها :

| | | | | | | | | | |
|--------------------------|---------------|--|-----------------|---|---------------------|---|----------------------|-----------------|-------------|
| <u>START</u> file - name | <u>KEY IS</u> | <table border="0"><tr><td><u>EQUAL TO</u></td></tr><tr><td>=</td></tr><tr><td><u>GREATER THAN</u></td></tr><tr><td>></td></tr><tr><td><u>NOT LESS THAN</u></td></tr><tr><td><u>NOT <</u></td></tr></table> | <u>EQUAL TO</u> | = | <u>GREATER THAN</u> | > | <u>NOT LESS THAN</u> | <u>NOT <</u> | data - name |
| <u>EQUAL TO</u> | | | | | | | | | |
| = | | | | | | | | | |
| <u>GREATER THAN</u> | | | | | | | | | |
| > | | | | | | | | | |
| <u>NOT LESS THAN</u> | | | | | | | | | |
| <u>NOT <</u> | | | | | | | | | |

[INVALID KEY imperative statement.]

حيث أن file - name هو اسم الملف الذي تجري فيه المعالجة ؛ ويجب أن تكون المعالجة SEQUENTIAL أو DYNAMIC ويجب أن يكون مفتوحاً بشكل INPUT أو I - O.

بعد تنفيذ هذه التعليمية وإذا لم يتحقق الشرط المطلوب نكون أمام حالة . INVALID KEY

مثال :

START VALFIL KEY IS > REC - KEY INVALID

KEY GO TO WRONG - RTN.

أما data - name فهي تشير الى حقل المفتاح للسجل أو للجزء اليساري منه والذي هو موصوف بمستوى أدنى مثل : RECI بالمثل التالي :

Ø3 REC - KEY.

Ø5 RECI PIC xx.

Ø5 REC 2 PIC xxx.

ح - تعليمة حذف سجل
DELETE
الشكل العام للتعليمة :

DELETE file - name RECORD.

تفيد هذه التعليمة في حذف سجل من الملف . و يشترط لتنفيذ هذه التعليمة أن يكون السجل قد قرء بتعليمة القراءة READ كما يجب أن يكون الملف مفتوحاً بشكل INPUT - OUTPUT أي I - O .

كما لاحظنا في تعليمة REWRITE السابقة ، فهنا أيضاً نجد أن هذه التعليمة توفر على المبرمج معالجة الملفات وعمل مقارنة بين ملف حركة وملف تعديل من أجل حذف سجل معين وكما مر معنا في المثال السابق .

ط - تعليمة CLOSE

هذه التعليمة لا تتغير مع هذا النوع من أساليب المعالجة و يبقى الشكل العام :

CLOSE file - name - 1 [,file - name - 2, ...] [WITH LOCK].

وأحياناً يضاف إليها عبارات خاصة بنوع وسط التخزين المستخدم مثل الاغلاق وعدم السماح لاحد بالتعديل على هذا الملف WITH LOCK وهذه العبارات تختلف باختلاف الحاسب المستخدم.

٨-٢٠٥-٣ معالجة الملفات المباشرة بالطريقة المتسلسلة *

يمكن أن نقول أن هذا الأسلوب من المعالجة للملفات المباشرة التنظيم DIRECT هي شبيهة بأسلوب معالجة الملفات المتسلسلة المفهرسة بالطريقة المتسلسلة مع بعض الاختلافات التي سنذكرها :

— في قسم ENVIRONMENT DIVISION نجد :

•• FILE - CONTROL.

SELECT file - name ASSIGN TO implementor - name

ORGANIZATION IS DIRECT

ACCESS MODE IS SEQUENTIAL,

RECORD KEY IS data - name - 1

[ALTERNATE RECORD KEY IS data - name - 2]

[FILE STATUS data - name - 3]

نجد أن الاختلاف فقط في عبارة ORGANIZATION DIRECT ، أي في أسلوب

بناء الملف - تنظيم الملف -

— وفي قسم PROCEDURE DIVISION :

تستخدم التعليمات :

REWRITE, WRITE, READ, OPEN, CLOSE, START, DELETE

(*) قليلة الاستخدام في لغة كوبرل على أجهزة غير أجهزة I.B.M

(••) T1990 COBOL MANUALS

كما هي مستخدمة في الملفات متسلسلة التنظيم والواردة في الفقرة ٨-٥-٢-١-٢ / أ ، ب ، ج ، د ، هـ ، ح ، ط ويمكن الرجوع إليها من أجل الشرح .

٨-٥-٢-١-٤ معالجة الملفات النسبية RELATIVE FILES بطريقة المعالجة المتسلسلة يعالج هذا النوع من الملفات النسبية حسب التسلسل المتصاعد ASCENDING لرقم السجل النسبي RELATIVE RECORD NUMBER لكل السجلات الموجودة في الملف .

و يكون التوصيف كالتالي :

— في قسم ENVIRONMENT DIVISION نجد :

FILE - CONTROL.

SELECT file - name, ASSIGN TO device - name

ORGANIZATION IS RELATIVE

ACCESS MODE IS SEQUENTIAL,

RELATIVE RECORD KEY IS data - name - 1

حيث أن أسلوب تنظيم الملف هونسيبي ORGANIZATION RELATIVE

وإن أسلوب المعالجة هو متسلسل ACCESS MODE SEQUENTIAL أما
RELATIVE RECORD KEY فهو رقم السجل النسبي وهو بمثابة مفتاح ، فيجب أن
يكون معرّفاً في DATA DIVISION لكن ليس من ضمن تعريف السجل في الملف أي
ليس في FD التابع لنفس الملف . يمكن أن يكون في فصل WORKING - STORAGE
ولا يجب أن يكون موصوفاً بشكل INDEX أو 1 - COMP أو 3 - COMP ولكن يجب أن
يعرف من النوع الرقمي (9) .

— في قسم PROCEDURE DIVISION :

نجد أن

أ- تعليمة فتح الملف OPEN :

OPEN { INPUT
OUTPUT
I - O } file - name الشكل العام لها

وهو نفس الشكل العام المستخدم في معالجة الملفات المتسلسلة.

ب- تعليمة READ :

كذلك أن الشكل العام للتعليمة هو نفس الاشكال السابقة والواردة في اسلوب
المعالجة التسلسلية

READ file - name NEXT RECORD [INTO identifier]
[AT END imperative statement]

ج- تعليمة WRITE :

نفس الشرح الوارد في الفقرة ٨-٥-٢-١-٢-ج
والشكل العام :

WRITE record name [FROM identifier] INVALID KEY
imperative statement.

د- تعليمة REWRITE :

أيضاً كما هو وارد في ٨-٥-٢-١-٢-د
والشكل العام :

REWRITE recordname [FROM identifier].

هـ- تعليمة START :

كما هي عليه في الفقرة ٨-٥-٢-١-٢ هـ

| | | | |
|--------------------------|---------------|----------------------|-------------|
| <u>START</u> file - name | <u>KEY IS</u> | <u>EQUAL</u> | data - name |
| | | = | |
| | | <u>GREATER THAN</u> | |
| | | > | |
| | | <u>NOT LESS THAN</u> | |
| | | <u>NOT</u> < | |

ح- تعليمة حذف سجل DELETE :

راجع الفقرة ٨-٥-٢-١-٢ ح

والشكل العام لها :

DELETE file - name RECORD.

ط- تعليمة غلق الملفات CLOSE :

والشكل العام لها :

CLOSE file - name - 1 [, file - name - 2 ...].

كما نلاحظ أنه يمكن معالجة الملفات النسبية بصورة بسيطة عند حذف سجل أو عند إضافة سجل جديد على الملف، أو حتى تعديل محتوى السجل. لكن يتم ذلك بطريقة تسلسلية والمبرمج لا يحدد مباشرة السجل الذي يريد العمل عليه إنما عليه القراءة تسلسلياً حتى يصل للرقم المطلوب ثم يجري عليه العمليات اللازمة.

كما ونلاحظ أيضاً أن الملف يعرف بفتح واحد فقط ولا توجد مفاتيح ثانوية

. ALTERNATE KEYS

٨-٥-٢-٢ المعالجة العشوائية (RANDOM ACCESS (DIRECT ACCESS)

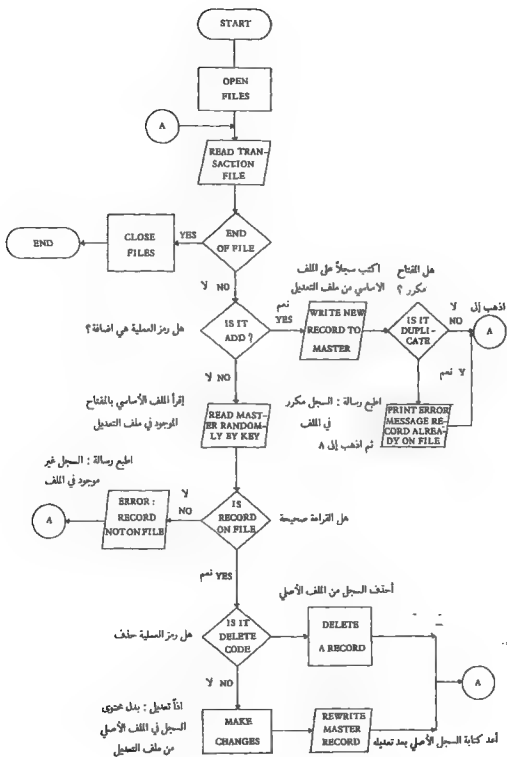
كما علمنا من الفقرات السابقة أن الملف الأساسي MASTER FILE بحاجة دوماً لتحديث UP - TO - DATE حتى تكون المعلومات فيه صحيحة مئة بالمئة.

إن هذه الطريقة من المعالجة والتي تعالج الملفات ذات التنظيم المفهرس INDEXED والمباشر DIRECT والنسبي RELATIVE تسمح بأن تعدل محتوى أي سجل بصورة بسيطة.

وتتم المعالجة غالباً باستخدام ملف تعديلات TRANSACTION FILE على الملف الأساسي MASTER FILE دون الحاجة للملف لإخراج جديد للملف الأساسي وكما مر معنا في الملفات المتسلسلة التنظيم.

إن ملف التعديل هذا يقرأ من الملف الأساسي بصورة عشوائية دون ترتيب معين و يعالج هذا السجل حسب رمز معين، هذه المعالجة تتضمن تعديل محتوى سجل موجود في الملف الأساسي أو حذف سجل موجود من الملف الأساسي (أحياناً الحذف يكون منطقياً LOGICAL بحيث يوضع بمكان معين في الملف الأساسي رمز يدل على أن هذا السجل محذوف) أو يمكن إضافة سجل جديد على الملف الأساسي بشرط أن يكون هذا السجل غير موجود فيه وإلا تكون عندنا حالة تكرار DUPLICATE وتعالج بعبارة ... INVALID KEY، والشكل التالي يرينا المخطط الانسيابي للمعالجة.

أنظر الصفحة رقم ٥٩٦ لتوصيف شكل المخرجات.



```

*
IDENTIFICATION DIVISION.
PROGRAM-ID. TXT20.
AUTHOR. KHAYAT-UNAID.
*
* يقوم هذا البرنامج بتعديل الملف الاساسي
* الملف الرئيسي من ملف تعديل متصل
*
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT INDXMST ASSIGN TO INDX
    ORGANIZATION INDEXED;
    ACCESS MODE RANDOM;
    RECORD KEY SUB-NO.
    SELECT TRANSF ASSIGN TO TRANS
    ORGANIZATION SEQUENTIAL
    ACCESS MODE SEQUENTIAL.
    SELECT PRINTF ASSIGN TO UT-S-PRFL.
DATA DIVISION.
FILE SECTION.
FD INDXMST LABEL RECORD IS STANDARD
DATA RECORD IS SUB-REC.
01 SUB-REC.
    03 SUB-NO PIC X(5).
    03 SUB-NAME PIC X(20).
    03 SUB-ADRES PIC X(24).
    03 SUB-DATE.
        05 SUB-YEAR PIC 9(4).
        05 SUB-MNTH PIC 99.
        05 SUB-DAY PIC 99.
    03 SUB-CODE PIC 99.
    03 SUB-AREA PIC 99.
    03 FILLER PIC X(19).
*
FD TRANSF LABEL RECORD IS STANDARD
BLOCK CONTAINS 0 RECORDS
DATA RECORD IS T-REC.

```

```

01  T-REC.
    03  T-NO      PIC  X(5).
    03  T-NAME    PIC  X(20).
    03  T-ADRES   PIC  X(24).
    03  T-DATE.
        05  T-YEAR  PIC  9(4).
        05  T-MNTH  PIC  99.
        05  T-DAY   PIC  99.
    03  T-CODEX   PIC  99.
    03  T-AREA    PIC  99.
    03  T-CODE     PIC  99.
    03  FILLER    PIC  X(17).

*
FD  PRINTF      LABEL RECORD IS OMITTED
                        DATA RECORD IS P-REC.
01  P-REC      PIC  X(132).

*
WORKING-STORAGE SECTION.
01  LN1.
    03  FILLER  PIC  X(47)  VALUE SPACES.
    03  FILLER  PIC  X(32)  VALUE
        'قائمة اخطاء تعديل الملف الاساسي'.
    03  FILLER  PIC  X(47)  VALUE SPACES.

*
01  LN2.
    03  FILLER  PIC  X(10)  VALUE SPACES.
    03  FILLER  PIC  X(20)  VALUE
        'رقم المشترك'.
    03  FILLER  PIC  X(20)  VALUE SPACES.
    03  FILLER  PIC  X(20)  VALUE
        'المشترك'.
    03  FILLER  PIC  X(30)  VALUE SPACES.
    03  FILLER  PIC  X(10)  VALUE
        'الملاحظات'.
    03  FILLER  PIC  X(16)  VALUE SPACES.

*
01  DETAILS.
    03  FILLER  PIC  X(20)  VALUE SPACES.
    03  P-NO    PIC  ZZZ99.
    03  FILLER  PIC  X(25)  VALUE SPACES.
    03  P-NAME  PIC  X(30).

```



```

*
  IF T-CODE = '01'
    PERFORM INSRT-RTN THRU EXT-I
  ELSE
    IF T-CODE = '02'
      PERFORM UPDAT-RTN THRU EXT-U
    ELSE
      IF T-CODE = '03'
        PERFORM DELET-RTN THRU EXT-D.

```

```

*
      GO TO READ-TRANS.

```

```

*
*   المذكوره التاليه هي لاضافه سجل جديد
*

```

```

  INSRT-RTN.
    PERFORM MOVE-TRNS-MST THRU EXT-MOVE.
    WRITE SUB-REC
      INVALID KEY MOVE
        'السجل مكرر' TO P-REMK
        PERFORM PRINT-LN GO TO EXT-I.
    ADD 1 TO ADD-CNTR.
  EXT-I.
    EXIT.

```

```

*
*   المذكوره التاليه هي لتعديل سجل من الملف الاساسي
*

```

```

  UPDAT-RTN.
    MOVE T-NO TO SUB-NO.
    READ INDXMST INVALID KEY
      MOVE 'السجل غير موجود' TO P-REMK
      PERFORM PRINT-LN GO TO EXT-U.
    PERFORM MOVE-TRNS-MST THRU EXT-MOVE.
    REWRITE SUB-REC INVALID KEY
      MOVE 'السجل غير موجود' TO P-REMK
      PERFORM PRINT-LN GO TO EXT-U.
    ADD 1 TO UPD-CNTR.
  EXT-U.
    EXIT.

```


*
* الذكرة التالية هي لمحتوى سجل من الملف الأساسي
*

```

DELET-RTN.
    MOVE T-NO TO SUB-NO.
    DELETE INDXMST RECORD INVALID KEY
        MOVE 'السجل غير موجود' TO P-REMK
        PERFORM PRINT-LN GO TO EXT-D.
    ADD 1 TO DEL-CNTR.
EXT-D.
    EXIT.

```

*
* الذكرة التالية هي لطباعة السجلات التي ليس لها
* أصل في الملف الأساسي
*

```

PRINT-LN.
    ADD 1 TO REF-CNTR.
    MOVE T-NO TO P-NO.
    MOVE T-NAME TO P-NAME.
    IF LN-CNTR > 24
        PERFORM N-PAGE THRU E-PAGE.
    MOVE SPACES TO P-REC.
    WRITE P-REC FROM DETAILS.
    WRITE P-REC FROM DASH.
    ADD 1 TO LN-CNTR.
EXT-LN.
    EXIT.

```

*
* الذكرة التالية هي لنقل الحظوظ من ملف الشركة
* إلى الملف الأساسي
*

```

MOVE--TRNS-MST.
    MOVE T-NO TO SUB-NO.
    MOVE T-NAME TO SUB-NAME.
    MOVE T-ADRES TO SUB-ADRES.
    MOVE T-DATE TO SUB-DATE.
    MOVE T-CODEX TO SUB-CODE.
    MOVE T-AREA TO SUB-AREA.
EXT-MOVE.
    EXIT.

```

*
* الذكرة التالية هي لقب المرحله وكتابه العناوين

*
N-PAGE.

```

MOVE SPACES TO      P-REC.
WRITE P-REC  AFTER  PAGE.
WRITE P-REC  FROM  LN1 AFTER 2 LINES.
WRITE P-REC  FROM  LN2 AFTER 3 LINES.
WRITE P-REC  FROM  AST AFTER 1 LINE.
MOVE SPACES TO      P-REC.
MOVE ZERO   TO      LN-CNTR.

```

E-PAGE.
EXIT.

*
* الذكرة التالية هي لطباعه المجاميع النهائية

*
END-OF-JOB.

```

PERFORM N-PAGE      THRU E-PAGE.
MOVE 'المضافه' TO    P-RST.
MOVE ADD-CNTR      TO    P-CNTR.
WRITE P-REC        FROM  FINLIN AFTER 20.
MOVE 'المعدله' TO    P-RST.
MOVE UPD-CNTR      TO    P-CNTR.
WRITE P-REC        FROM  FINLIN AFTER 2.
MOVE 'المحذوفه' TO    P-RST.
MOVE DEL-CNTR      TO    P-CNTR.
WRITE P-REC        FROM  FINLIN AFTER 2.
MOVE 'المرفوضه' TO    P-RST.
MOVE REF-CNTR      TO    P-CNTR.
WRITE P-REC        FROM  FINLIN AFTER 2.
WRITE P-REC        FROM  AST.

```

*
CLOSE INDXMST, TRANSF, PRINTF.

EXT-END.
EXIT.

*

ملاحظة : إن حاصل التعديل بهذه الطريقة هو مساوي ومطابق تماماً لحاصل التعديل في المعالجة المتسلسلة إنما طريقة العمل مختلفة .

على أن هذه ليست الطريقة الوحيدة في المعالجة ؛ انما يمكن التعديل أن يكون مباشرة من الشاشة الى الملف دون الحاجة الى ملف تعديل ؛ وفيه تظهر شاشة أولية يعطى فيها خيار إما لحذف أو لاضافة أو لتعديل سجل ، بعدها يطلب رقم السجل المراد العمل عليه ويدخل عن طريق الشاشة ويستدعى الى الشاشة و يظهر محتواه ثم تنفذ العملية المطلوبة ، اما في حالة الاضافة فتدخل البيانات مباشرة من الشاشة الى الملف الموجود على قرص مخفض .

ويمكن تتبع المخطط المنطقي السابق في المعالجة عن طريق الشاشة إنما عوضاً عن القراءة من ملف تعديل يطلب القراءة من الشاشة فقط .

سنستعرض بالتفصيل المعالجة العشوائية لكل نوع من أنواع تنظيم الملفات :

٨-٥-٢-١ معالجة الملفات المتسلسلة بالطريقة العشوائية :

لا يمكن معالجة الملفات ذات التنظيم التسلسل SEQUENTIAL ORGANIZATION بالطريقة العشوائية RANDOM ACCESS وذلك لعدم وجود علاقة بين مفتاح السجل ووسط التخزين .

٨-٥-٢-٢ معالجة الملفات المفهرسة المتسلسلة بالطريقة العشوائية :

ENVIRONMENT DIVISION نجد أنه في قسم
:
FILE - CONTROL.

SELECT implementor - name ASSIGN TO

ORGANIZATION IS INDEXED,

ACCESS MODE IS RANDOM,

RECORD KEY IS data - name - 1,

[ALTERNATE KEY IS data - name - 2, [WITH DUPLICATES]

[FILE STATUS IS data - name - 3].

مفتاح السجل : RECORD KEY

يجب تعيين اسم الحقل الذي يعتبر مفتاحاً. واسم الحقل هذا يجب أن يرد توصيفه في توصيف الملف في DATA DIVISION، و يعتبر هذا المفتاح هو الذي يميز سجلاً عن آخر ولا يجوز أن يكون مكرراً في الملف و يسمى PRIME KEY أي المفتاح الأولي. كما لا يجوز تعديل هذا المفتاح بتغيير الرقم، وإن اضطر المبرمج لأن يعدل المفتاح فما عليه إلا أن يحذف السجل ثم يضيفه برقم جديد.

المفتاح الملحق : *ALTERNATE KEY

هذه العبارة تعرف المفتاح الثانوي، الثالثي، الرابعي... للملف؛ إذ يمكننا أن نتعرف على السجل بأكثر من مفتاح، فمثلاً سجل الموظف يمكن أن نتعرف عليه برقمه كمفتاح أساسي أولي أو باسم الموظف كمفتاح ثانوي، أو بمرتبه أي مفتاح ثالث أو برقم حقيقته وهو مفتاح رابع....

فإذا كان من الممكن أن يتكرر المفتاح الثانوي أو الثالثي... في الملف كأن يكون أكثر من موظف له نفس الاسم الثلاثي، أو أن يكون أكثر من موظف بنفس المرتبة فنضيف عبارة - WITH DUPLICATES - وهي تعرف الحاسب أن هذا المفتاح غير الأولي مكرر في الملف، أما إذا لم نذكر هذه العبارة WITH DUPLICATES أمام توصيف المفتاح الثانوي فهذا يعني أن هذا المفتاح الثانوي وحيد أيضاً.

(هـ) يمكن أن يصل عدد المفاتيح الثانوية إلى ثلاثة عشر مفتاحاً في حاسب T1990 وتختلف باختلاف الحاسب المستخدم.

أما عبارة حالة الملف FILE STATUS فتشير الى حالة الملف ويجب أن نحجز مساحة في WORKING-STORAGE حتى نعرف 3 - name - data ويجب أن تكون رقمية وبطول حرفين (2 NUMERIC BYTES)؛ إن هذا الحقل يتبدل حسب تبدل حالة الملف فيوضع فيه أرقام لها معنى في كل حاسب، لذلك يرجع الى مراجع كل حاسب من أجل معرفة معنى هذه الأرقام.

فلو كانت القراءة ناجحة مثلاً لكان محتواه صفراً؛ أما في حالة عدم وجود المفتاح في الملف فيتوضع فيه قيمة يدل المرجع على أن السجل غير موجود. وهكذا بالنسبة لكل معالجة تطرأ على الملف.

— في قسم الاجراءات PROCEDURE DIVISION

نستعرض فيما يلي التعليمات والشكل العام لكل منها مع شرح لها :

أ. تعليمية OPEN :

OPEN { INPUT file - name - 1 ...
OUTPUT file - name - 3 ...
I - O file - name - 4 ... }

كما مر معنا في الشرح السابق الفقرة ٨ - ٥ - ٢ - ١ - ٢ فقرة أ.

ب. تعليمية القراءة READ :

الشكل العام لها :

READ file - name RECORD [INTO identifier] [KEY IS data - name]

[INVALID KEY imperative - statement].

تستخدم هذه التعليمات من أجل القراءة من ملف ذي تنظيم مفهرس وبمعرفة المفتاح data - name وهو نفسه الموصف في FD للملف، وكذلك في SELECT ومحتوى هذا المفتاح يأتي إما عن طريق ملف آخر مسلسل sequential هو ملف تعديل مثلاً أو عن طريق الشاشة بتعليمات ACCEPT .

وفي حالة نجاح القراءة - كان السجل موجوداً - ينتقل التنفيذ للتعليمات التالية للـ READ وإلا يذهب التنفيذ الى التعليمات التي بعد عبارة INVALID KEY .

مثال :

READ TRNSF AT END GO TO E - O - JOB.

MOVE T - NO TO MST - NO.

READ MSTFILE RECORD KEY IS MST - NO

INVALID KEY GO TO ERR - RTN.

MOVE

ADD

نلاحظ في المثال : أنه في البداية قرأنا من ملف التعديل بصورة متسلسلة (لأن تنظيمه متسلسل مثلاً) ثم حركنا محتوى المفتاح T - NO الموجود في ملف التعديل والذي نريد أن تعدل نظيره في الملف الأساسي؛ نحرك محتواه الى مفتاح الملف الأساسي MST - NO ، ونقرأ الملف الأساسي MSTFILE بصورة عشوائية RANDOM (دون أي ترتيب أو مقارنة - انظر المعالجة المتسلسلة) فإذا تمت القراءة بنجاح انتقل التنفيذ للتعليمات التالية للـ READ وهي MOVE ثم ADD ، أما إذا كان هذا الرقم T - NO غير موجود في الملف الأساسي فتتشكل حالة INVALID KEY والتنفيذ يذهب الى التعليمات GO TO ERR - RTN وهناك يظهر أو يطبع رسالة حسب المطلوب .

مثال ٢ :

ACPT.

ACCEPT MST - NO.

READ MSTFILE RECORD KEY

IS MST - NO INVALID KEY

DISPLAY "RECORD IS NOT EXIST"

GO TO ACPT.

DISPLAY "RECORD IS EXIST"

⋮

نلاحظ في المثال الثاني : ان محتويات المفتاح قد أخذناه عن طريق الشاشة بتعليمة ACCEPT ، ونقرأ فيه الملف ؛ فاذا كانت القراءة ناجحة أي الملف موجود ظهرت عبارة RECORD IS EXIST ، أما اذا كانت القراءة غير ناجحة فننشأ حالة INVALID KEY وتظهر رسالة "RECORD IS NOT EXIST" وتذهب المعالجة الى ACPT لقبول سجل جديد ، وهكذا

WRITE تعليمة الكتابة

الشكل العام :

WRITE record - name [FROM identifier] [INVALID KEY

imperative - statement].

تستخدم كما اسلفنا في اساليب المعالجة التسلسلة والعشوائية للملفات ذات التنظيم المفهرس والعشوائي والنسبي ، هذه التعليمة تكتب على ملف المعلومات المنقولة من ملف آخر أو عن طريق الشاشة .

وحالة INVALID KEY تنشأ هنا عندما يكون محتوى حقل المفتاح RECORD KEY له مماثل في الملف الاساسي المراد الكتابة عليه ، أي حالة تكرار

المفتاح على نفس الملف. ويجب ان تنفذ هذه التعليمة بعد فتح الملف كملف اخراج
 OUTPUT أو ملف ادخال - اخراج I - O كما يجب أن نذكر أنه في حال وجود مفتاح
 ثانوي وغير معرف بعبارة WITH DUPLICATES أي لا يقبل التكرار وأضفنا سجل
 مفتاحه الاساسي غير مكرر، أما الثانوي فهو مكرر لكن دون ذكر العبارة السابقة
 WITH DUPLICATES فإن حالة INVALID KEY تتشكل أيضاً، لذلك يجب
 الانتباه ليس للمفتاح الاساسي فقط بل والثانوي أيضاً.

مثال :

WRITE ENDX - REC INVALID KEY GO TO ERR - RTN.

PERFORM TOP - DOWN THRU EXT - TOP.

⋮

في هذا المثال نجد أنه اذا نفذت تعليمة WRITE بنجاح ينتقل تسلسل التنفيذ الى
 تعليمة PERFORM وإلا تتشكل حالة INVALID KEY و ينتقل التنفيذ الى تعليمة
 GO TO ERR - RTN وفيها يعالج السجل حسب المطلوب.

د. تعليمة اعادة الكتابة REWRITE ;
 الشكل العام لها :

REWRITE record - name [FROM identifier]

[INVALID KEY imperative - statement].

تستخدم هذه التعليمة لتثبيت تعديل سجل معين؛ وتتشكل حالة
 INVALID KEY عندما يكون محتوى المفتاح الأولي PRIME KEY غير مطابقة لأي
 قيمة في الملف الأساسي - السجل غير موجود -

مثال :

READ - LOOP.

READ TFILE AT END GO TO E - O - PROCESS.

MOVE T - NO TO MST - NO.

READ MSTFILE INVALID KEY GO TO PRT - ERR.

MOVE T - NAME TO MST - NAME.

MOVE T - ADRESS TO MST - ADDRESS.

↓

REWRITE MST - REC INVALID KEY GO TO PRT - ERR.

GO TO READ - LOOP.

PRT - ERR.

نلاحظ من المثال السابق أنه تمت القراءة من ملف التعديل ثم حُرِّك المفتاح من ملف التعديل الى مفتاح الملف الأساسي.

بعدها تمت القراءة من الملف الأساسي في حال عدم نجاح القراءة يذهب التنفيذ الى فقرة PRT - ERR وإلا يتابع عمله بتحريك الحقول من ملف التعديل الى حقول الملف الأساسي بعدها يُثبت التعديل بتعليمة إعادة كتابة السجل REWRITE، فإذا كان محتوى المفتاح الاساسي قد غير أثناء المعالج فتتشكل حالة INVALID KEY وإلا يتابع التنفيذ الى تعليمة GO TO READ - LOOP ... وهكذا.

هـ- تعليمة : START

لا تنفذ مع اسلوب المعالجة العشوائية.

ح - تعليمية الحذف DELETE : الشكل العام :

DELETE file - name [RECORD] [INVALID KEY imperative - statement].

تستخدم هذه التعليمية لحذف سجل من الملف ويمكن استخدام مكان هذا السجل لإضافة سجل آخر، ويمكن للسجل أن يحذف منطقياً LOGICALLY أي نظرياً، لكن بالواقع يكون موجود ولكن لا يمكن معالجته، وفي كلا الحالتين يمكن للحاسب استخدام مكانه لتسجيل سجل آخر.

ويجب أن ننوه أنه من مقتضيات المعالجة أحياناً يلزم أن نحذف سجلاً - نوقف معالجته - لفترة معينة ثم نعاود إضافته للملف ومعالجته، فإذا استخدمنا تعليمية DELETE فنجد أن محتوى السجل قد فقد كلياً؛ لذا نستخدم حقلاً معيناً ضمن السجل ونعطيه رمزاً ونضع في هذا الحقل رمز الحذف وبالتالي نقارن عليه فإذا كان رمز الحذف موجوداً يعني أنه محذوف مؤقتاً.

مثال :

لواخذنا رمز الحذف المؤقت هو T والحذف الدائم هو D في ملف التعديل وحقل تثبيت الحذف في الملف الأسامي هو STS - MST يكون :

READ TFILE AT END GO TO E - O - JB.

MOVE T - NO TO MST - NO.

IF T - STS = "D"

حذف حقيقي
DELETE MST - FILE RECORD INVALID KEY

PHISICAL DELTION

GO TO ERR - RTN.

MOVE T - NO TO MST - NO.

READ MSTFILE INVALID KEY GO TO ERR - RTN.

MOVE "T" TO MST - STS.

حذف نظري REWRITE MST - REC INVALID KEY GO TO ERR - RTN.

LOGICAL DELETION

تتم قراءة ملف التعديل، فإذا كان الرمز D يعني أنه مطلوب حذف السجل نهائياً من الملف الأساسي لذلك تحرك قيمة المفتاح ويحذف بموجبه السجل نهائياً من الملف بتعليمة DELETE أما إذا لم يكن الرمز D وكان T فهذا يعني أنه حذف مؤقت فيحرك المفتاح ثم يقرأ السجل المطلوب حذفه حذف مؤقت تحرك قيمة T (دالة على الحذف المؤقت) إلى الملف الأساسي و يعاد كتابة السجل من جديد بتعليمة REWRITE، وفي أثناء المعالجة يعمل تحقق على الحقل MST - STS فإذا كانت التعليمة T موجودة فهذا يعني أنه محذوف مؤقتاً فلا يعالج، أما إذا كان غير ذلك فيعالج.

وعند الانتهاء من توقيف هذا السجل عن المعالجة يحرك للحقل MST - STS القيمة SPACE و يعاد كتابة السجل من جديد بتعليمة REWRITE فيرجع السجل كما كان في السابق ويكون تحت المعالجة.

ط - تعليمة CLOSE :

لها الشكل العام CLOSE file - name - 1, ...

٨-٥-٢-٣ معالجة الملفات المباشرة بالطريقة العشوائية :

نجد أنه يمكن معالجة الملفات ذات التنظيم المباشر DIRECT بالطريقة العشوائية

RANDOM وذلك حسب الوصف التالي :

— في قسم ENVIRONMENT DIVISION :

FILE - CONTROL.

SELECT file name ASSIGN TO ...

ORGANIZATION IS DIRECT,
ACCESS MODE IS RANDOM
RECORD KEY IS data - name - 1,
[ALTERNATE RECORD KEY IS data - name - 2]
[WITH DUPLICATES]
[FILE STATUS IS data - name - 4].

يرجع للفقرة ٨ - ٥ - ٢ - ٢ - ٢ من أجل شرح عبارة RECORD KEY ،
FILE STATUS ، WITH DUPLICATES ، ALTERNATE RECORD KEY
والاختلاف الوحيد هو في عبارة ORGANIZATION IS DIRECT أما بالنسبة
للتعليمات فهي كما وردت في الفقرة ٨ - ٥ - ٢ - ٢ - ٢ أ ، ب ، ج ، د ، هـ ، ح ،
ط .

والاختلاف في المعالجة هو ضمنى ويخص الحاسب إذ يتم البحث عن السجل في
هذه الطريقة من المعالجة بواسطة عملية حسابية معقدة RANDOMIZATION ويصل
للعنوان المطلوب و يقرأ السجل فإذا لم يجده تتشكل حالة INVALID KEY بينما في
المعالجة المفهرسة يتم البحث عن طريق ملف ملحق يدعى الفهرس INDEX و يأخذ
العنوان منه ليقرأ ملف البيانات .

كما قلنا ان طريقة المعالجة ضمنية ولا تهم المبرمج في لغة كويول في شيء .

٨ - ٥ - ٢ - ٢ - ٤ معالجة الملفات النسبية بالطريقة العشوائية
كما مر معنا سابقاً إن الملفات النسبية ذات ترتيب متسلسل بالنسبة لبداية الملف ؛
فان التوصيف يكون على الشكل

— في قسم ENVIRONMENT DIVISION :

• FILE - CONTROL.

SELECT file ASSIGN TO ...
ORGANIZATION IS RELATIVE
ACCESS MODE IS RANDOM,
RELATIVE KEY IS data - name - 1
[FILE STATUS IS data - name - 2.]

كما نلاحظ هنا فإنه لا يوجد لدينا RECORD KEY إنما يستعاض عنها بـ
RELATIVE KEY بالمفتاح النسبي وهو رقم السجل ضمن الملف ولا يكون موصفاً في
FD التابع للملف file name نفسه.

كما أنه لا يوجد هنا ALTERNATE KEY مفتاح ثانوي ، لأن أسلوب التنظيم هنا
بأن يعرف كل سجل حسب تسلسل موصفه بالنسبة لبداية الملف .
إن تسلسل المعالجة هنا يحدده المبرمج فهو الذي يطلب سجلاً معيناً للتعديل أو
الحذف وذلك بوضع قيمة المفتاح في الحقل المسمى RELATIVE KEY بعد ذلك تتم
المعالجة وفق هذه القيمة .

— في قسم الاجراءات PROCEDURE DIVISION

نجد أن الكلام الذي ينطبق على معالجة الملفات ذات التنظيم الم فهرس بالطريقة
العشوائية ينطبق على أسلوب المعالجة للملفات ذات التنظيم النسبي وذلك للتعليمات
الواردة في الفقرة ٨-٥-٢-٢-٢ ضمن أ ، ب ، ج ، د ، ح ، ط لذلك يرجع
لتلك التعليمات من أجل معرفة استخدامها .

عدا تعليمية START التي لا تستخدم مع اسلوب المعالجة العشوائية
.RANDOM ACCESS

انظر التمارين الملحقه في هذا الفصل من أجل استخدام أنواع اساليب المعالجة
المختلفة.

٨ - ٥ - ٢ - ٣ المعالجة الحركية DYNAMIC ACCESS

في كثير من الأحيان يضطر المبرمج لأن يستخدم كلاً من اسلوبي المعالجة المتسلسلة
والعشوائية في برنامج واحد لمعالجة ملف واحد لذلك أوجدت طريقة المعالجة الحركية
والتي فيها يستطيع المبرمج أن يعالج ملفاً معيناً باسلوبي معالجة متسلسلة وعشوائية.
والفكرة تتضح أكثر بالمثال التالي :

لوفرضنا أنه لدينا ملف يحتوي على اسماء طلاب الجامعات في المملكة ولوفرضنا
أنه لدينا ١٠ جامعات وكل جامعة فيها عدد معين من الكليات وفي كل كلية هناك
نظام السنوات وكل سنة عدد من الطلاب . فلوطلب منا معلومات معينة عن الجامعة
رقم ٨ مثلاً ؛ فلو اتبعت الطريقة المتسلسلة في المعالجة SEQUENTIAL ACCESS للزم
الأمر أن نقرأ جميع طلاب الجامعات السبع حتى نصل للجامعة رقم ٨ لبدء المعالجة وهذا
يستغرق وقتاً طويلاً .

ولو اتبعنا طريقة المعالجة العشوائية بالوصول للجامعة رقم ٨ لما أمكننا متابعة العمل
على هذه الجامعة لأن هذا الأسلوب يوصلنا لأول سجل في الجامعة رقم ٨ ولا نستطيع
أن نتابع المعالجة من مكاننا هذا .

هنا نجد أن استخدام اسلوب المعالجة الحركي يسمح للمبرمج أن يصل لسجل معين
وأن يبدأ منه المعالجة المتسلسلة ففي مثالنا السابق يمكن أن نصل الى الجامعة رقم ٨
مباشرة بتعليمية START وأن نبدأ المعالجة منها بتعليمية READ NEXT حتى ننتهي
من جميع طلاب الجامعة المطلوبة ثم ننتقل لغيرها وهكذا

واسلوب المعالجة الحركي يمكن أن يأتي مع كل أنواع تنظيم الملفات ، ويمكن لجميع التعليمات السابقة أن تستخدم فيه ، لذلك يرجع للفقرة ٨ - ٥ - ٢ من أجل استخدام التعليمات المختلفة وهي :

- أ $\text{OPEN} \left\{ \begin{array}{l} \underline{\text{INPUT}} \\ \underline{\text{OUTPUT}} \\ \underline{\text{I - O}} \end{array} \right\} \begin{array}{l} \text{file - name - 1,} \\ \text{file - name - 2,} \\ \text{file - name - 3, ...} \end{array}$
- ب $\left\{ \begin{array}{l} \underline{\text{READ}} \text{ file - name} \\ \underline{\text{READ}} \text{ file - name} \end{array} \right\} \begin{array}{l} \underline{\text{NEXT RECORD AT END}} \text{ imperative - statement} \\ \underline{\text{INVALID KEY}} \text{ imperative - statement} \end{array}$
- ج $\left\{ \begin{array}{l} \underline{\text{WRITE}} \text{ Record - name.} \\ \underline{\text{WRITE}} \text{ Record - name} \end{array} \right\} \underline{\text{INVALID KEY}} \text{ imperative - statement}$
- د $\underline{\text{START}} \text{ file - name } \underline{\text{KEY IS}} \left\{ \begin{array}{l} \underline{\text{EQUAL TO}} \\ \underline{\text{GREATER THAN}} \quad \text{data - name} \\ \underline{\text{NOT LESS THAN}} \end{array} \right\}$
- [$\underline{\text{INVALID KEY}} \text{ imperative - statement}$]
- هـ $\underline{\text{REWRITE}} \text{ record - name } [\underline{\text{INVALID KEY}} \text{ imperative - statement}]$
- و $\underline{\text{DELETE}} \text{ file - name } \underline{\text{RECORD}} [\underline{\text{INVALID KEY}} \text{ imperative - statement}]$
- ط $\underline{\text{CLOSE}} \text{ file - name, ... } [\underline{\text{WITH LOCK}}]$.

و يكون شكل SELECT هو التالي :

SELECT file - name ASSIGN TO implementor - name,

ORGANIZATION IS $\left\{ \begin{array}{l} \underline{\text{SEQUENTIAL}} \\ \underline{\text{INDEXED}} \\ \underline{\text{DIRECT}} \\ \underline{\text{RELATIVE}} \end{array} \right\}$

ACCESS MODE IS DYNAMIC,

الجدول التالي يبين لنا التعليمات المسموح استخدامها حسب نوع المعالجة، ونوع الملف المستخدم.

| | نوع المعالجة | التعليمات | Open Mode | |
|-----------------------|--------------|-----------|-----------|--------|
| | | | ادخال | اخراج |
| | File Access | Statement | INPUT | OUTPUT |
| I - O | | | | |
| * Mode | | | | |
| Sequential متسلسلة | | READ | x | |
| | | WRITE | | x |
| | | REWRITE | | |
| | | START | x | |
| | | DELETE | | |
| Random عشوائية | | READ | x | |
| | | WRITE | | x |
| | | REWRITE | | |
| | | START | | |
| | | DELETE | | |
| Dynamic حركية | | READ | x | |
| | | WRITE | | x |
| | | REWRITE | | |
| | | START | x | |
| | | DELETE | | |

تظهر علامة x التعليمة المسموح استخدامها مع نوع فتح الملف مثال :
 في نوع المعالجة العشوائية RANDOM تعليمة READ مثلاً تستخدم عند فتح الملف
 أما ك INPUT أو O - I أو ادخال - اخراج .
 نلاحظ أن في نوع المعالجة الحركية DYNAMIC أن كل التعليمات يمكن أن
 تستخدم عند فتح الملف ك ادخال اخراج O - I .

٨ - ٦ اسئلة وتمارين

سؤال ١ - لدينا ملف للزبائن اسمه CUSTFILE يحتوي على سجلات CUSTREC
 والسجل يشمل رقم الزبون CUSTNO ويمثل مفتاح السجل KEY وبيانات أخرى عن
 الزبون مثل اسمه وعنوانه وحالته المالية . فإذا أردنا كتابة برنامج يساعد في معرفة
 الزبون الجديد إذا كان موجود سابقاً أولاً . وإذا لم يكن موجوداً إضافة سجل جديد
 على الملف ، فما هي العبارات والتعليمات التي يجب ذكرها في البرنامج والتي تشمل
 النواحي التالية علماً بأن الملف منظم كملف مفهرس ، وفق رقم الزبون .

أ - عبارة SELECT لتداول الملف CUSTFILE
 ب - العبارات أو التعليمات اللازمة لإضافة زبون جديد على الملف مع الإشارة الى
 هذه الإضافة على وحدات الاخراج (شاشة أو طباعة) .

سؤال ٢ - لدينا الملف DATEFILE الموصوف في قسم البيانات كما يلي :

FD DATEFILE ...

01 DATREC

03 YEAR - MONTH.

05 YY PIC XX.

5 MM PIC 99.

3 ETC - FLD PIC X (76).

وهذا الملف مفهرس حسب الحقل YEAR - MONTH ، والمطلوب :

أ - كتابة التعليمات اللازمة لقراءة هذا الملف اعتباراً من السنة رقم 79 ثم نقل التسلسل الى الفقرة END - PROCESS عندما تتبدل السنة 79 .

ب - ما هي صيغة التداول ACCESS التي سوف توصف في عبارة SELECT .

سؤال ٣ - أشر الى العبارة الصحيحة بـ (ـ) وإلى الخطأ بـ (x) لكل من العبارات التالية :

١ - يُنشأ عادة الملف المفهرس Indexed file مع طريقة تداول عشوائية ACCESS IS RANDOM ولكنه يُقرأ مع طريقة تداول متسلسلة . ACCESS IS SEQUENTIAL

٢ - إذا وردت العبارة ACCESS IS SEQUENTIAL مع عبارة SELECT عندئذ يمكن معالجة الملف الموصوف بطريقة عشوائية Randomly اذا كان سجله يحتوي على مفتاح KEY .

٣ - تعليمة REWRITE يمكن أن تأتي فقط مع الملف الذي تم فتحه على أساس I-O .

٤ - يمكن استخدام جملة INVALID KEY مع تعليمة READ أو تعليمة WRITE .

٥ - إن الملفات المخزنة على الأقراص المغنطة لا يمكن معالجتها تسلسلياً Sequentially أبداً .

٦ - يرد مع تعليمة DELETE اسم السجل المراد حذفه من الملف .

٧ - تستخدم تعليمة SET لأجل تخزين قيمة الدلائل Index وتحديد قيمته .

- ٨ - يرد تعريف ملف الفرز في قسم البيانات DATA DIVISION وفي الفصل الخاص بالملفات FILE SECTION تحت اسم SD .
- ٩ - إن الملف الموصوف كملف مفرس Indexed لا يحتاج الى عملية فرز SORT على المفاتيح (المفتاح) المعرقة في عبارة SELECT لأنه يُرتب آلياً .
- ١٠ - حذف سجل من الملف النسبي RELATIVE بتعليمة DELETE يؤدي الى وجود فراغ مكان السجل المحذوف وبالتالي يمكن كتابة سجل آخر جديد يحمل نفس رقم المفتاح مكانه .

تمرين ١ :

لدينا ملفان موجودان على شريط ممغنط مفروزان تصاعدياً على رقم السلعة؛ الملف الأول يحتوي على سجلات ذات أرقام السلع المفردة، والملف الثاني يحتوي على سجلات ذات أرقام السلع المزدوجة؛ المطلوب منك كتابة برنامج يدمج ملفي الإدخال في ملف إخراج واحد اسمه OUTMRG علماً بأن شكل السجلات كالتالي :

الملف الأول (الأرقام المفردة) : INFILE 1

IN1 - NO PIC 9 (5)

IN1 - DATA PIC X (35)

الملف الثاني (الأرقام المزدوجة) : INFILE 2

IN2 - NO PIC 9 (5)

IN2 - DATA PIC X (35)

ملف الإخراج : OUTMRG

OUT - NO PIC 9 (5)

OUT - DATA PIC X (35)

مفتاح الحل :

- ١ - أقرأ السجلين الأول والثاني .
- ٢ - قارن مفتاحي السجلين IN1 - NO و IN2 - NO وانتقل سجل الرقم الأصغر الى ملف الإخراج واكتبه .
- ٣ - أعد القراءة من الملف الذي يحتوي على الرقم الأصغر - المنقول سابقاً - .
- ٤ - استخدم HIGH - VALUES من أجل معالجة نهايات الملفات .

الفصل التاسع

البرمجة الهيكلية بلغة كوبول

٩ - ١ لمحة عامة

يكون هدف المبرمج المبتدئ بلغة كوبول كتابة برنامج بسيط يعمل بدون خطأ ويمكن أن يسمى Working Program ولكن بعد فترة من الممارسة يصبح الغرض الأول للمبرمج كتابة برنامج فعال وكفؤ يمكن تسميته Efficient Program وتناز هذا النوع الآخر بسهولة قراءته ومراجعته وإمكانية تطويره وتعديله من قبل مبرمج آخر غير المبرمج الذي قام بكتابته. وبذلك يكون الهدف من البرمجة بلغة كوبول باختصار: برنامجاً سهل القراءة والمراجعة وسليماً ومختبراً بشكل جيد من الناحية المنطقية ليؤدي الغرض الذي كتب من أجله.

وعلى مر الزمن بُدئ في تجميع أساليب تقنية وطرق من أجل التوصل إلى طريقة لكتابة برنامج يحقق الأهداف المذكورة آنفاً. وسوف نقدم فقط ضمن هذا المؤلف فكرة عن البرمجة الهيكلية Structured Programming كوسيلة مساعدة في لغة كوبول للوصول إلى برنامج سهل القراءة وذو منطق سليم. وكذلك سنتعرض إلى نقاط قياسية أثبتت الخبرة الطويلة في برمجة كوبول وجوب وجودها في البرنامج السليم المنشود. وهذه النقاط تساعد على الوصول إلى برنامج سهل القراءة والمراجعة والتطوير في المستقبل دون تعقيد، وليست ملزمة للمبرمج المبتدئ.

٩ - ٢ نظرية البرمجة الهيكلية STRUCTURED PROGRAMMING

قد يكون المبرمج يتبع في برنامجه عناصر البرمجة الهيكلية المنطقية منذ مدة ولكنه لم يلتفت الى تصنيفها على النحو الذي وضعت به هذه الهياكل المنطقية في نطاق نظرية تدعى نظرية البرمجة الهيكلية والتي تهدف الى وضع نظام من أجل كتابة البرنامج السهل والمنطقي. وقد احتوت هذه النظرية على ثلاثة هياكل منطقية أساسية وهي: التسلسل Sequence والاختيار Selection والدوارة Iteration كما هومبين في الأشكال الموضحة برسم المخطط الانسيابي Flowchart **.

وهيكل التسلسل يؤكد على تنفيذ التعليمات ضمن البرنامج بشكل متتالي، وهذا هو الشكل الطبيعي لأي برنامج ما لم يتم نقل التسلسل من قبل المبرمج. فمثلا العملية A يجب أن تنفذ قبل B كما هو في المخطط الانسيابي.

أما هيكل الاختيار فهو عبارة عن مقارنة بين فعلين أو عمليتين واختبار شرط تحقق أحدهما يسمى Predicate أو Test. فإذا كانت نتيجة الشرط صحيحة True يتم تنفيذ العملية A وإذا كانت النتيجة غير صحيحة False يتم تنفيذ العملية B.

أما الدوارة فهي عبارة عن تكرار تنفيذ عملية معينة عدد من المرات طالما أن شرط نهايتها صحيح True وإذا أصبح هذا الشرط غير صحيح فتكون العملية قد توقفت عن التكرار. فالعملية A سوف تتكرر عدداً من المرات طالما أن الشرط P صحيح. وعندما يصبح هذا الشرط غير صحيح False تتوقف تنفيذ هذه العملية. وينتقل البرنامج الى العملية أو التعليمة التالية.

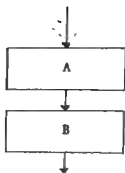
و يلاحظ من أشكال البرمجة الهيكلية أنها تأخذ مدخلا واحداً One Entry ومخرجا واحداً One Exit لكل شكل من عناصرها، وهذا أمر ضروري لسهولة تتبع خطوات البرنامج.

(*) Techniques of Program Structure and Design, Edward Yourdon, Prentice - Hall, 1975, PP 145 - 147.

(**) هناك عناصر ثانوية أخرى ورموز خاصة بهذه الطريقة لا سبيل لذكرها في هذا المؤلف.

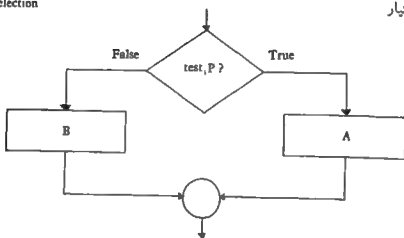
1. Sequence

١ - التسلسل



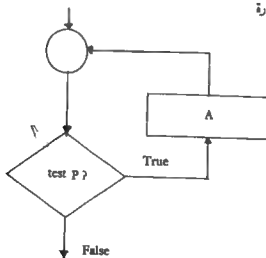
2. Selection

٢ - الاختيار



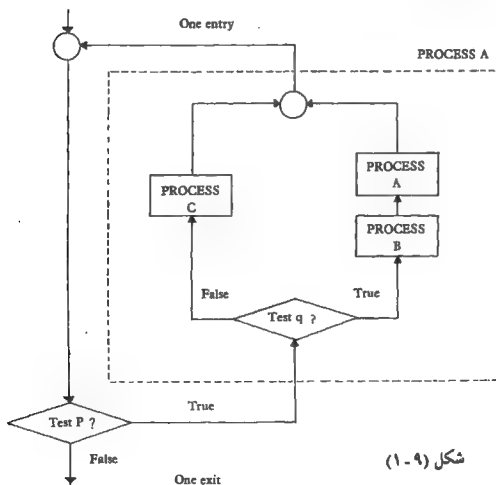
3. Iteration

٣ - الدوارة



كما أن هذه الأشكال الأساسية يمكن أن تجتمع في برنامج واحد بحيث تكون العملية A تحتوي على تسلسل واختيار كما هو موضح في الرسم التالي لمخطط انسيابي. (الشكل رقم ٩-١).

ويلاحظ أن هذه الأشكال لا تحتوي على ما يسمى بلغة كوبرول بتعليمة GO TO ليس لأن هدف البرمجة الهيكلية كتابة برامج بدون GO TO بل لأنها تزيد في صعوبة تتبع البرنامج ومعرفة خطواته المنطقية وهذا يتعارض مع سهولة القراءة والفهم المطلوبين في البرنامج المقصود باستخدام أسلوب البرمجة الهيكلية.



شكل (٩-١)

٩- ٣ أشكال البرمجة الهيكلية بلغة كوبول :

إن هيكل التسلسل مطبق أصلا في لغة كوبول حيث أن الوضع الطبيعي لتنفيذ تعليمات البرنامج على التوالي . أما هيكل الاختيار والمسمى IFTHENELSE ينطبق بوساطة تعليمة IF ... ELSE وكذلك هيكل الدوارة والمسمى DOWHILE ينطبق بتعليمة PERFORM ... UNTIL .

الاختيار : ويكتب في شكله العام :

IF Condition

Statement - 1

ELSE

Statement - 2.

حيث أن Statement - 1 و Statement - 2 يمكن أن تكون تعليمة واحدة من لغة كوبول أو مجموعة تعليمات .

مثال :

IF STD - CLG NOT = PRVUS - CLG

PERFORM PRINT - YEAR THRU EXIT - CLG

ELSE

PERFORM PRINT - RECORD THRU EXIT - PRIN.

وإذا كان هناك أكثر من مقارنة فيمكن أن نأخذ الشكل التالي كمثال بلغة كوبول :

IF STD - CLG NOT = PRVUS - CLG

PERFORM PRINT - YEAR THRU EXIT - CLG

ELSE

IF STD - YEAR NOT = PRVUS - YEAR

PERFORM PRINT - YEAR THRU EXIT - YEAR

ELSE

PERFORM PRINT - RECORD.

ويجب ملاحظة وجود هامش في السطر الثاني لتعليمة IF وكلمة ELSE وكذلك كتابة كلمة IF وكلمة ELSE الخاصة بها على نفس المستوى العمودي لسهولة القراءة والفهم.

الدوارة :

تأخذ الشكل العام

PERFORM Procedure - name

UNTIL Condition - P (= true)

ومثال على ذلك التعليمات التالية :

MOVE 'NO' TO END - OF - JOB.

PERFORM READ - RECORD

UNTIL END - OF - JOB = 'YES'.

READ - RECORD.

READ STD - FILE

AT END MOVE 'YES' TO END - OF - JOB.

NEXT - PARAGRAPH.

٩- ٤ ارشادات في ترميز برنامج كوبول

على ضوء البرمجة الهيكلية

الى جانب استخدام اسلوب البرمجة الهيكلية في وضع المخطط المنطقي لهيكل البرنامج يجب على المبرمج بلغة كوبول تتبع الارشادات التي تساعد في الوصول الى برنامج سهل وسليم المنطق وعلى الرغم من أنها لا تعتبر ضمن قواعد لغة كوبول . ولكنها تكتسب بالخبرة العملية والممارسة الطويلة مع هذه اللغة من البرمجة وقد تختلف من مركز حاسب الى آخر في تطبيقها .

قسم البيانات : DATA DIVISION

- ١ - ابتداء جميع عبارات PICTURE من نفس العمود وقد يكون عادة العمود ٤٠ تقريباً أو حوله . وعلى المبرمج توحيد الكلمة في البرنامج المستخدم وهي إما PIC أو PIC IS أو PICTURE وهذا ينطبق على عبارة VALUE وعبارة USAGE .
- ٢ - استخدام رمزين من أول الحقول الموصوفة في السجل الواحد للملف الواحد لاجل تمييزها بسهولة . مثل

ST - NAME, ST - ADDRESS, ST - TELEPHONE

- ٣ - حذف ادخالات المستوى رقم 77 ومحاولة تجميعها حسب نوعيتها ضمن المستوى رقم 1٠.

- ٤ - ترك هوامش متسلسلة في أرقام المستوى عند توصيف السجل في الملف . كذلك استخدام أرقام مستوى بينها فجوات رقمية لأجل السماح بالاضافة بالمستقبل

مثل ٥, 1٠, 15, 2٠,

أو 3, ٧, ٩, 11, 15

- ٥ - استخدام أسماء حقول ذات دلالة ومعنى وليس مجرد حروف ورموز مبهمه مثل ST - X بدلاً من ST - CODE

وكذلك يجب التنويه الى عدم الافراط في الايضاح والتركيز على اختصار الاسم مع المحافظة على المعنى والدلالة .

٦ - استخدام هامش عند كتابة السطر الواحد على سطرين لعدم كفاية السطر الأول ، من أجل الإشارة الى استمرارية السطر الأول مثل :

05 FILLER PIC X (13) VALUE

' ---- BOOK ----'

والشكل التالي يلخص ما سبق التنويه إليه :

01 STUDENT - IN - REC.

05 ST - NAME.

10 ST - FIRST - NAME PIC X (25).

10 ST - LAST - NAME PIC X (25).

05 ST - BIRTH - DATE.

10 ST - BIRTH - DAY PIC 99.

10 ST - BIRTH - MONTH PIC 99.

10 ST - BIRTH - YEAR PIC 99.

WORKING - STORAGE SECTION.

01 FLAGS

05 TOTAL - FLAG PIC 9.

05 AGE - FLAG PIC 9.

قسم الاجراءات PROCEDURE DIVISION

- ١ - تجنب كتابة أكثر من تعليمة على نفس السطر، وذلك من أجل سهولة المراجعة والقراءة رغم أن مترجم كوبول يقبل أكثر من تعليمة على نفس السطر.
- ٢ - يفضل وضع أسماء الفصول والفقرات على أسطر مستقلة، وكذلك محاولة ترك سطر فارغ قبل هذه الأسماء لزيادة الوضوح والقراءة السهلة.
- ٣ - إذا كان هناك أسماء فقرات أو فصول في قسم الاجراءات متسلسلة فيجب أن تكون أرقامها متسلسلة أيضاً وتحتوي على فجوات رقمية لامكانية حشر فقرات أخرى عند التعديل إذا اضطر لذلك المبرمج. مثل

0010 - HOUSEKEEPING.

0020 - READ - REC.

- ٤ - ترك فراغات مناسبة لإبراز التعليمات المهمة في البرنامج وكذلك استخدام الهوامش لهذا الغرض عند وجود تعليمة IF المتداخلة.
- ٥ - ربط تعليمة IF مع كلمة ELSE الخاصة بها ووضعها في نفس المستوى عمودياً عند الكتابة. لأن كل ELSE ترجع الى أقرب IF في حال وجود IF المتداخلة.
- ٦ - تنظيم أسماء الحقول في التعليمات المتشابهة عمودياً مثل تعليمة MOVE كما في المثال التالي :

MOVE ST - NAME TO P - NAME.

MOVE ST - ADRESS TO P - ADRESS.

MOVE ST - SCORES TO P - SCORES.

- ٧ - صياغة التعليمات بطريقة سهلة دون تعقيد خاصة في التعليمات الحسابية لئلا يسبب التحويل في العمليات الحسابية صعوبة في فهم الترميز بلغة كوبول وعلى الأخص عند استخدام تعليمة COMPUTE.

والى جانب هذه الارشادات في القسمين يوجد بعض النقاط المشتركة بينهما يمكن
إيجازها فيما يلي :

قسم البيانات وقسم الاجراءات :

- ١- يفضل استخدام أسطر فارغة أثناء الترجمة وعند الكتابة فمثلا الرمز / يقوم بقلب الصفحة عند الترجمة عند وضعه في العمود ٧ أو استخدام أوامر SKIP EJECT * وترك أسطر فارغة بين التعليمات لابرار المهم منها.
- ٢- الاستفادة من الأعمدة ٧٣ الى ٨٠ في صفحة الكتابة بلغة كوبول لغرض الإشارة الى التعديلات الواردة على البرنامج بعد كتابته فمثلا لو تم وضع :
BKH 05/82
- للدلالة على أوائل أحرف من اسم المبرمج الذي قام بالتعديلات وهي BKH و 05 تدل على الشهر في سنة ١٩٨٢ وهو التاريخ الذي تم تعديل هذا السطر في البرنامج .
- ٣- استخدام الملاحظات المناسبة بين أسطر البرنامج للإشارة الى وظيفة بعض الأجزاء أو الفقرات فيه ، مع الأخذ بعين الاعتبار عدم الافراط في ذلك .
- ٤- تجنب استخدام الفواصل Commas و يكفي استخدام أعمدة فارغة بين الجمل أو العبارات التي لا تحتاج الى وجود نقطة في نهايتها ، وذلك لكي لا يختلط شكل الفاصلة مع النقطة على المبرمج .

٩-٥- مثال عملي :

في الصفحات التالية نموذج عن برنامج بلغة كوبول قد صمم ورمز مع مراعاة البرمجة الهيكلية .

(٥) موزودة فقط في أنظمة تشغيل IBM

```

*
IDENTIFICATION DIVISION.
PROGRAM-ID. TXT21.
AUTHOR. KHAYAT-JNAID.
*
* يقوم هذا البرنامج بطباعة المجاميع
* الجزيئية والمتوسطة والنتيجه
*
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT STDFILE ASSIGN TO UT-S-STDf.
    SELECT PRINTF ASSIGN TO UT-S-PRFL.
DATA DIVISION.
FILE SECTION.
FD STDFILE LABEL RECORD IS STANDARD
                                DATA RECORD IS STD-REC.
01 STD-REC.
    03 STD-NO PIC 9(5).
    03 STD-NAME PIC X(30).
    03 STD-CLG PIC 99.
    03 STD-YEAR PIC 99.
    03 STD-DATA PIC X(11).
*
FD PRINTF LABEL RECORD IS OMITTED
                                DATA RECORD IS P-REC.
01 P-REC PIC X(132).
*
WORKING-STORAGE SECTION.
01 HDR-1.
    03 FILLER PIC X(52) VALUE SPACES.
    03 FILLER PIC X(28) VALUE
        'KING SAUD UNIVERSITY '.
    03 FILLER PIC X(52) VALUE SPACES.
01 HDR-2.
    03 FILLER PIC X(11) VALUE 'DATE '.
    03 DAY-P PIC 99.
    03 FILLER PIC X VALUE '/'.

```

| | | | | |
|----|--------------------|-----|-------|------------------|
| 03 | MONTH-P | PIC | 99. | |
| 03 | FILLER | PIC | XXX | VALUE '/19'. |
| 03 | YEAR-P | PIC | 99. | |
| 03 | FILLER | PIC | X(31) | VALUE SPACES. |
| 03 | FILLER | PIC | X(15) | VALUE |
| | 'STUDENTS REPORT'. | | | |
| 03 | FILLER | PIC | X(34) | VALUE SPACES. |
| 03 | FILLER | PIC | X(06) | VALUE 'PAGE : '. |
| 03 | P-CTR | PIC | ZZ9. | |
| 03 | FILLER | PIC | X(11) | VALUE SPACES. |
| 01 | HDR-3. | | | |
| 03 | FILLER | PIC | X(10) | VALUE SPACES. |
| 03 | FILLER | PIC | X(17) | VALUE |
| | 'COLLEGE CODE'. | | | |
| 03 | FILLER | PIC | X(19) | VALUE |
| | 'YEAR NO '. | | | |
| 03 | FILLER | PIC | X(16) | VALUE |
| | 'STUDENT NUMBER'. | | | |
| 03 | FILLER | PIC | X(16) | VALUE SPACES. |
| 03 | FILLER | PIC | X(14) | VALUE |
| | 'STUDENT NAME '. | | | |

*

| | | | | |
|----|----------|-----|---------|---------------|
| 01 | DETAILS. | | | |
| 03 | FILLER | PIC | X(16) | VALUE SPACES. |
| 03 | P-CLG | PIC | 99. | |
| 03 | FILLER | PIC | X(12) | VALUE SPACES. |
| 03 | P-YEAR | PIC | 99. | |
| 03 | FILLER | PIC | X(18) | VALUE SPACES. |
| 03 | P-NO | PIC | Z(3)99. | |
| 03 | FILLER | PIC | X(17) | VALUE SPACES. |
| 03 | P-NAME | PIC | X(30). | |
| 03 | FILLER | PIC | X(15) | VALUE SPACES. |

*

| | | | | |
|----|------------------------|-----|-------|---------------|
| 01 | TOTL-YEAR. | | | |
| 03 | FILLER | PIC | X(23) | VALUE |
| | ' TOTAL OF YEAR NO. '. | | | |
| 03 | P-TYEAR | PIC | 99. | |
| 03 | FILLER | PIC | X(15) | VALUE ' : '. |
| 03 | P-CYEAR | PIC | ZZ99. | |
| 03 | FILLER | PIC | X(90) | VALUE SPACES. |


```

*
01  TOTL-CLG.
03  FILLER PIC      X(29)  VALUE
    '  ** TOTAL OF COLLEGE CODE '.
03  P-TCLG PIC      99.
03  FILLER PIC      X(5)   VALUE ' : '.
03  P-CCLG PIC      ZZZ99.
03  FILLER PIC      X(90)  VALUE SPACES.

```

```

*
01  TOTL-OV-ALL.
03  FILLER PIC      X(24)  VALUE SPACES.
03  FILLER PIC      X(32)  VALUE
    '  TOTAL OF ALL THE STUDENTS : '.
03  P-OVAL PIC      ZZZ999.
03  FILLER PIC      X(60)  VALUE SPACES.

```

```

*  لقبول التاريخ من الحاسب
01  CURENT-DATE.
03  C-YEAR PIC      99.
03  C-MONTH PIC     99.
03  C-DAY  PIC      99.

```

```

*  أعدادات تستخدم للتجميع
01  TOTAL-COUNTERS.
03  YEAR-CNTR PIC   9(4).
03  COLG-CNTR PIC   9(4).
03  OVAL-CNTR PIC   9(6).

```

```

*  أعدادات تستخدم للصفحة والسطر
01  LINE-PAGE-COUNTERS.
03  LN-CNTR  PIC   99.
03  PG-CNTR  PIC   999.

```

```

*  مساعدات عمل تستخدم لمقارنته السجل الحالي
*  مع السابق بعد ان نكون قد حفظنا فيها
*  القيم السابقة للمقول المعينه بالمعالجه
01  GLD-VALUES.
03  PRVS-YEAR PIC   99.
03  PRVS-CLG  PIC   99.

```

```

*
77  END-OF-JB PIC   XXX.

```

```

*
PROCEDURE DIVISION.
    OPEN      INPUT      STDFILE
              OUTPUT     PRINTF.
* قبول التاريخ من المناسب وطباعه
*
    ACCEPT    CURENT-DATE FROM DATE.
    MOVE      C-YEAR      TO   YEAR-P.
    MOVE      C-MONTH     TO   MONTH-P.
    MOVE      C-DAY       TO   DAY-P.
    PERFORM   NEW-PAGE     THRU  EXTP.

* التعليمات الخمس التاليه تشكل المعالجه
* الرئيسه في البرنامج
    PERFORM   READ-FILE    THRU   EXIT-READ.
    MOVE      STD-CLG      TO   PRVS-CLG P-TYEAR.
    MOVE      STD-YEAR     TO   PRVS-YEAR P-TCLG.
    PERFORM   MAIN-LOOP    THRU   EXIT-LOOP
              UNTIL         END-OF-JB = 'YES'.
    PERFORM   END-JOB      THRU   EXIT-END.
    STOP      RUN.

*
MAIN-LOOP.
    IF      STD-CLG      NOT EQUAL PRVS-CLG
        PERFORM PRINT-YEAR THRU EXIT-CLG
    ELSE
    IF      STD-YEAR     NOT EQUAL PRVS-YEAR
        PERFORM PRINT-YEAR THRU EXIT-YEAR
    ELSE
        PERFORM READ-FILE THRU EXIT-READ.
EXIT-LOOP.
EXIT.

*
* قراءه سجل من الملف لمعالجته
*
READ-FILE.
    READ STDFILE AT END
    MOVE 'YES' TO END-OF-JB.
EXIT-READ.
EXIT.

```

```

*
* : طباعه سجل درسي
*
PRINT-RECORD.
    MOVE STD-NO TO P-NO.
    MOVE STD-NAME TO P-NAME.
    MOVE STD-CLG TO P-CLG.
    MOVE STD-YEAR TO P-YEAR.
    WRITE P-REC FROM DETAILS
    AFTER 2.
    ADD 1 TO LN-CNTR.
    IF LN-CNTR > 18
    PERFORM NEW-PAGE THRU EXTP.
    ADD 1 TO YEAR-CNTR.
EXIT-PRINT.
EXIT.

```

```

*
* : طباعه مجاميع الطلاب لسنه معينه
*
PRINT-YEAR.
    MOVE YEAR-CNTR TO P-CYEAR.
    WRITE P-REC FROM TOTL-YEAR
    AFTER 2.
    ADD YEAR-CNTR TO COLG-CNTR.
    MOVE STD-YEAR TO PRVS-YEAR
    P-TYEAR.
EXIT-YEAR.
EXIT.

```

```

*
* : طباعه مجاميع الطلاب لكليه معينه
*
PRINT-CLG.
    MOVE COLG-CNTR TO P-CCLG.
    WRITE P-REC FROM TOTL-CLG
    AFTER 2.
    ADD COLG-CNTR TO OVAL-CNTR.
    MOVE STD-CLG TO PRVS-CLG
    P-TCLG.
EXIT-CLG.
EXIT.

```

```

*
* ذكره قلب المصنفه وكتابه العناوين:
*
NEW-PAGE.
    MOVE ZERO TO LN-CNTR.
    ADD 1 TO PG-CNTR.
    MOVE PG-CNTR TO P-CTR.
    MOVE SPACES TO P-REC.
    WRITE P-REC AFTER PAGE.
    WRITE P-REC FROM HDR-1
    AFTER 1.
    WRITE P-REC FROM HDR-2
    AFTER 3.
    WRITE P-REC FROM HDR-3
    AFTER 2.
EXTP.
EXIT.
*
* ذكره طباعه المجاميع النسابيه
*
END-JOB.
    PERFORM PRINT-YEAR THRU EXIT-CLG.
    MOVE OVAL-CNTR TO P-OVAL.
    WRITE P-REC FROM TOTL-OV-ALL
    AFTER 2.
    CLOSE STDFILE
    PRINTF.
EXIT-END.
EXIT.

```

٩ - ٦ اسئلة وتمارين

سؤال ١ - إذا كان لدينا قسم الاجراءات التالي :

PROCEDURE DIVISION.

MAIN - P SECTION.

PERFORM FIRST - RTN.

PERFORM SECOND - RTN.

PERFORM THIRD - RTN UNTIL TOT - SUM 12.

STOP RUN.

NEXT - P SECTION.

FIRST - RTN.

MOVE 1 TO N.

MOVE ZEROS TO TOT - SUM.

SECOND - RTN.

COMPUTE $M = N * 2$.

COMPUTE $X = N * 3$.

ADD 1 TO N.

ADD M, X, N, GIVING SUM - 1.

THIRD - RTN.

ADD SUM - 1 TO TOT - SUM.

FOURTH - RTN.

ADD 1 TO N.

ADD 2 TO X.

NEXT - Q SECTION.

.....

المطلوب :

أ - ما هي القيمة النهائية للحقل SUM - 1 ؟

ب - ما هي القيمة النهائية للحقل TOT - SUM ؟

ج - كم مرة تُنفذ الفقرة RTN - THIRD ؟

د - هل هناك فقرات لا تنفذ وما هي ؟

هـ - هل هناك فقرات تُنفذ أكثر من مرة وما هي ؟

سؤال - ٢ في السطور التالية برنامج مكتوب بلغة كوبول اسمه NDXFILE يقوم بعملية إنشاء ملف مفهرس INDEXED من بطاقات مثقبة لبيانات عن الدارسين .
والمطلوب تتبع هذا البرنامج وملء الفراغات بالشكل الصحيح .

000110 IDENTIFICATION DIVISION.

000120 PROGRAM - 1D. NDXFILE.

000130 AUTHOR. AUTHORS.

000140** THIS PROGRAM IS TO DEMONSTRATE.

000150** INDEXED FILE CREATION.

000160 ENVIRONMENT DIVISION.

000170 CONFIGURATION SECTION.

000180 SOURCE - COMPUTER. IBM - 3033.

000190 OBJECT - COMPUTER. IBM - 3033.

000191 INPUT - OUTPUT SECTION.

000192 ← (1)

000193 SELECT CARD - FILE ASSIGN TO UT - S - SYSIN.
 000194 SELECT DISK - FILE ASSIGN TO VSAMFILE
 000195 ORGANIZATION IS INDEXED
 000196 ACCESS MODE IS ← (2)
 000197 RECORD KEY IS STUDENT - NUMBER.
 000198 SELECT PRINT - FILE ASSIGN TO UT - S - PRF.
 000199 DATA DIVISION.
 000200 FILE SECTION.
 000201 FD DISK - FILE LABEL RECORD IS ← (3)
 000202 RECORD CONTAINS 80 CHARACTERS
 000203 DATA RECORD IS DISK - REC.
 000204 01 DISK - REC.
 000205 05 STUDENT - NUMBER PIC ← (4)
 000206 05 STUDENT - DATA PIC X (60).
 000207 05 FILLER PIC X (12).
 000208 FD CARD - FILE LABEL RECORD IS ← (5)
 000209 DATA RECORD IS CARD - REC.
 000210 01 CARD - REC.
 000211 05 STUDENT - IDENT PIC 9 (8).
 000212 05 STUDENT - DATA PIC X (60).
 000213 05 FILLER PIC X (12).
 000215 FD PRINT - FILE LABEL RECORD IS OMITTED
 000216 DATA RECORD IS PRINT - REC.
 000217 01 PRINT - REC PIC X (132).
 000218*

000219 WORKING - STORAGE SECTION.
 000220 01 END - OF - DATA PIC X VALUE 'N'.
 000221 88 INPUT - END VALUE 'Y'.
 000222
 000223 PROCEDURE DIVISION.
 000224 MAINROUTINE.
 000225 OPEN INPUT CARD - FILE (6)
 000226 PRINT - FILE, DISK - FILE.
 000227 PERFORM READ - CARD.
 000228 PERFORM FILE - CREATE UNTIL INPUT - END.
 000229 CLOSE CARD - FILE, , PRINT - FILE. (7)
 000230 STOP RUN.
 000231
 000249 FILE - CREATE.
 000250 MOVE STUDENT - IDENT TO STUDENT - NUMBER.
 000260 MOVE STUDENT - DATA OF TO (8)
 000270 STUDENT - DATA OF (9)
 000280 WRITE DISK - REC
 000290 INVALID KEY ERROR - CASE. (10)
 000300 PERFORM READ - CARD.
 000400 READ - CARD.
 000500 READ CARD - FILE RECORD (11)
 000600 AT END MOVE TO END - OF - DATA.
 000700 ERROR - CASE.
 000800 MOVE 'RECORD EXIST IN THIS FILE'

000900 TO PRINT - REC.
001000 WRITE PRINT - REC AFTER ADVANCING 1 LINE. (12)
001100 WRITE FROM CARD - REC AFTER ADVANCING 1 LINE.
001200 EXIT - PAR. EXIT.

الفصل العاشر

تدقيق البرنامج واختباره

١٠-١ ملحة عامة

من المتوقع جداً وجود أخطاء في برنامج كوبرول خاصة تلك البرامج التي تكتب في المراحل الأولى من تعلم هذه اللغة . ويمكن تصنيف الأخطاء الى نوعين :

- ١ - أخطاء شكلية وتتعلق بقواعد اللغة واستخدام الكلمات الخاصة بها . وتظهر هذه الأخطاء أثناء عملية الترجمة `ERRORS in COMPILATION` .
- ٢ - أخطاء منطقية وتظهر هذه الأخطاء بعد خلو البرنامج من الأخطاء الشكلية أولاً ، ويكتشفها المبرمج بعد وضع البرنامج موضع التنفيذ (أو الاختبار) . ويمكن حصر هذه الأخطاء في حالتين :

- أ - إمكانية تنفيذ البرنامج ولكن يتم الحصول على نتائج غير سليمة .
- ب - توقف تنفيذ البرنامج بشكل غير طبيعي عند بعض فقرات أو تعليمات البرنامج .

١٠-٢ الأخطاء الشكلية `ERRORS in COMPILATION`

إن كل مترجم كوبرول يقوم بعملية فحص وتدقيق شكل السطر أو التعليمة المكتوبة وفق القواعد والأسس العامة لهذه اللغة وعند اكتشافه لأي عيب أو خطأ في هذه القواعد يعط إشارة خطأ على هذا السطر وأحياناً مع الإشارة الى مكان الخطأ في

نفس السطره وتصنف هذه الأخطاء الى نوعين بشكل عام :

- أخطاء أساسية و يرمز لها عادة بالحرف E ولا يمكن متابعة تنفيذ البرنامج بدون تصحيحها.
- أخطاء مميتة * و يرمز لها عادة بالحرف D وعند ظهورها في البرنامج يتوقف المترجم عن العمل بعدها .
- أخطاء تحذيرية و يرمز لها بالحرف W وفي بعض الأجهزة بالحرف C ، حيث يقبلها المترجم ولكن يعطي إشارة تحذيرية للمبرمج على مكان وجودها . ويمكن متابعة تنفيذ البرنامج بدون تصحيحها .
- أخطاء شرطية *** و يرمز لها بالرمز C و يفترض المترجم الحالة الصحيحة منها ويمكن للمبرمج متابعة تنفيذ البرنامج بدون تعديلها .

ويجب لفت نظر المبرمج الى أن المترجم قد يزيد الطين بلة عند اكتشافه لخطأ شكلي في أول البرنامج حيث أنه يعطي نفس الإشارة للدلالة على الخطأ برقم جديد عند كل استخدام لهذا الحقل أو الملف أو السجل في الأقسام التالية من البرنامج ، وأحياناً يكون عدد الأخطاء الشكلية Syntax بالئات في نهاية عملية الترجمة إلا أن السبب أو الخطأ الرئيسي يكون قليلاً جداً ولكن موقع هذا الخطأ يكون في مستهل البرنامج .

٣.١٠ الأخطاء المنطقية ERRORS IN EXECUTION

هذه الأخطاء تظهر عند اختبار البرنامج و بعد خلوه من أخطاء الشكل أو القواعد . وعند تدقيق النتائج الناجمة من عملية الاختبار يكتشف المبرمج هذه الأخطاء ، وهناك

(٥) تختلف باختلاف المترجم المستخدم

(**) Crawford and Grauer P . 72

(***) Conditional ويمكن أن تصنف مع الأخطاء التحذيرية في بعض المترجمات غير الخاصة بشركة IBM .

حالتان لظهور هذه الأخطاء :

- أ - انتهاء البرنامج المنفذ بشكل طبيعي مع الحصول على نتائج غير سليمة . وهذه الأخطاء في النتائج يمكن تفسيرها على شكلين .
- إما خطأ في سير منطق العمليات ضمن البرنامج وبشكل مخالف للمخطط الانسيابي Flowchart الموضوع لهذا البرنامج .
- أو خطأ في تطابق البيانات الداخلة الى البرنامج مع الشكل Format الموضوع لوصف هذه البيانات المدخلة .
وبالتالي يجب على المبرمج التأكد من تطابق وصحة البيانات المقروءة الى البرنامج من الملفات المخصصة لإدخال المعلومات ومعالجتها .
- ب - توقف البرنامج عن التنفيذ وانتهائه بشكل غير طبيعي أو دخوله في دوارة لا نهاية لها Endless Loop . وهذا ينجم عن خطأ منطقي في نقل المخطط الانسيابي الى رموز البرنامج أو أحياناً عن استخدام غير سليم للتعليمات الموضوعة للغة كوبرول .

١٠ - ٤ أخطاء شائعة

- هناك أخطاء شائعة الحدوث في لغة كوبرول يمكن تلخيصها في النقاط التالية لتساعد المبرمج في تحديد الخطأ الحاصل في البرنامج :
- ١ - اغفال تحديد نقطة البداية في الحقل أو المؤشر Subscript .
 - ٢ - وضع بيانات غير رقمية في حقول موصوفة رقمياً .
 - ٣ - الاستخدام غير الصحيح للنقطة في نهاية الجملة بلغة كوبرول .
 - ٤ - خروج قيمة المؤشر أو الدليل عن مجال الجدول الموضوع لأجله .

- ٥ - التداخل غير الصحيح في تعليمة PERFORM .
- ٦ - معالجة سجل ادخال مرتين .
- ٧ - عدم تطابق صيغة البيانات في الحجم والشكل الموصوفين في السجل Record .

١٠ - ٥ كشف الأخطاء وتصحيحها

يجب على المبرمج قبل الوقوع في الأخطاء من مختلف الأنواع مراعاة النقاط التالية أثناء تصميم برامجه للحصول على أدنى نسبة من الخطأ :

- تجنب التعليمات Instructions غير الواضحة والمبهمة مثل تعليمة المقارنة IF مع العوامل المنطقية AND , OR , NOT .
- الاستفادة قدر الامكان من اسلوب البرمجة الهيكلية Structured Programming (الفصل رقم ٩) وكذلك استخدام اسلوب التصميم الجزئ Modular Programming من أعلى الى اسفل عند وضع الخطوات الأساسية للبرنامج .
- اتباع اسلوب برمجة سهل وخالي من التعقيد وبالتالي خالي من الأخطاء .
- وفي حال حدوث الأخطاء يمكن للمبرمج اللجوء الى الطرق التالية من أجل اكتشافها وبالتالي تصحيحها ثم إعادة اختبار البرنامج .
- تدقيق البرنامج من قبل المبرمج نفسه للكشف عن الخطأ، والتحري عن الخطأ البسيط والأكثر احتمالاً ثم البحث عن الخطأ المعقد . ويستعان باستخدام الشك (الظن) في مواقع الخطأ المحتملة، ووضع نقاط علامة في البرنامج .
- الاستعانة بميزة أو خاصية كشف الخطأ في لغة كوبرول COBOL DEBUGGING وهذه الميزة تعطي المبرمج امكانية متابعة البرنامج

خطوة خطوة أثناء تنفيذه وبالتالي معرفة مواقع الخطأ .

- استخدام أوامر نصح (تفريغ) للبرنامج أثناء التنفيذ و يدعى الأمر DUMP وبموجبه يمكن للمبرمج من تفريغ محتويات الذاكرة أثناء تنفيذ البرنامج ، وعند توقف البرنامج غير الطبيعي أو عند حصول الخطأ الذي صعب كشفه بالطرق السابقة ، وغالباً ما تكون أشكال DUMP مكتوبة بالرموز السادسية عشر Hexadecimal وتبين فيها عناوين تخزين فقرات البرنامج ، وتختلف من حاسب لآخر لذلك ينصح بالرجوع الى مراجع الشركة المختصة ، لمعرفة امكانية قراءة هذه الكشوف الناجمة عن عملية التفريغ .

١٠-٦ تمارين

- ١ - في الصفحات التالية برنامج كويول و يظهر بعد ترجمته بواسطة مترجم نظام IBM وتبدو الأخطاء الناجمة عن عملية الترجمة Compilation في الصفحة الأخيرة من البرنامج وقد تم ذكر رقم السطر الذي يحتوي هذا الخطأ والمطلوب كشف الأخطاء وتصحيحها ليصبح البرنامج خالياً من أي منها .

(هـ) تكون متوفرة في بعض المترجمات الخاصة بلغة كويول . وللمزيد من المعلومات ينصح بالرجوع الى مراجع الشركة . أو الى الكتب المتقدمة في لغة كويول .

```

001 *
002 IDENTIFICATION DIVISION.
003 PROGRAM--ID. TESTP.
004 AUTHOR. KHAYAT--JNAID.
005 *
006 *
007 *
008 *
009 *
010 ENVIRONMENT DIVISION.
011 CONFIGURATION SECTION.
012 SOURCE--COMPUTER. IBM-370.
013 OBJECT--COMPUTER. IBM-370.
014 INPUT-OUTPUT SECTION.
015 FILE CONTROL.
016 SELECT STDCKF ASSIGN TO UT-S-INF1.
017 SELECT SRTFIL ASSIGN TO UT-S-INF3.
018 SELECT OUTSRT ASSIGN TO UT-S-MRGF.
019 SELECT PRINTF ASSIGN TO UT-S-PRFL.
020 DATA DIVISION.
021 FILE SECTION.
022 *
023 FD STDCKF LABEL RECORD IS STANDARD
024 RECORD CONTAINS 80 CHARACTERS
025 DATA RECORD IS STK-REC.
026 01 STK-REC PIC X(80).
027 *
028 SD SRTFIL
029 RECORD CONTAINS 80 CHARACTERS
030 DATA RECORD IS SRT-REC.
031 *
032 *
033 01 SRT-REC.
034 03 SRT-NO PIC 9(6).
035 03 SRT-DATA PIC X(74).
036 *
037 FD OUTSRT LABEL RECORD STANDARD
038 RECORD CONTAINS 80 CHARACTERS
039 DATA RECORD IS OUT-REC.
040 01 OUT-REC.
041 03 OUT-NO PIC 9(5).

```


| | | | | | |
|-----|----|--------------------------|-----|-------------------------|---------------|
| 042 | 03 | OUT-DESC | PIC | X(20). | |
| 043 | 03 | OUT-QTY | PIC | 9(5). | |
| 044 | 03 | OUT-CODE | PIC | 9(3). | |
| 045 | 03 | OUT-PRC | PIC | 9(3)V99. | |
| 046 | 03 | OUT-MIN | PIC | 9(5). | |
| 047 | 03 | FILLER | PIC | X(37). | |
| 048 | * | | | | |
| 049 | FD | PRINTF | | LABEL RECORD IS OMITTED | |
| 050 | | | | DATA RECORD IS P-REC. | |
| 051 | 01 | P-REC | PIC | X(132). | |
| 052 | * | | | | |
| 053 | * | | | | |
| 054 | | WORKING-STORAGE SECTION. | | | |
| 055 | 01 | HDR1. | | | |
| 056 | 03 | FILLER | PIC | X(57) | VALUE SPACES. |
| 057 | 03 | FILLER | PIC | X(17) | VALUE |
| 058 | | | | 'S T O C K | L I S T'. |
| 059 | 03 | FILLER | PIC | X(56) | VALUE SPACES. |
| 060 | * | | | | |
| 061 | 01 | HDR2. | | | |
| 062 | 03 | FILLER | PIC | X(08) | VALUE |
| 063 | | | | | ' DATE : '. |
| 064 | 03 | P-DATE | PIC | 99/99/99. | |
| 065 | 03 | FILLER | PIC | X(100) | VALUE SPACES. |
| 066 | 03 | FILLER | PIC | X(07) | VALUE |
| 067 | | | | | 'PAGE : '. |
| 068 | 03 | P-PAGE | PIC | ZZ9. | |
| 069 | 03 | FILLER | PIC | X(5) | VALUE SPACES. |
| 070 | * | | | | |
| 071 | 01 | HDR3. | | | |
| 072 | 03 | FILLER | PIC | X(40) | VALUE |
| 073 | | ' ITEM NO. | | ITEM | DESCRIPTION'. |
| 074 | 03 | FILLER | PIC | X(29) | VALUE |
| 075 | | ' QUANTITY | | ITEM | CODE'. |
| 076 | 03 | FILLER | PIC | X(17) | VALUE |
| 077 | | ' ITEM PRICE'. | | | |
| 078 | 03 | FILLER | PIC | X(46) | VALUE |
| 079 | | ' MINIMUM LEVEL | | | COMMENTS'. |
| 080 | * | | | | |
| 081 | 01 | DATA-LINE. | | | |
| 082 | 03 | FILLER | PIC | X(06) | VALUE SPACES. |

| | | | | | |
|-----|----|---------------------|----------------------|---------|---------------|
| 083 | 03 | P-NO | PIC | ZZZZ99. | |
| 084 | 03 | FILLER | PIC | X(06) | VALUE SPACES. |
| 085 | 03 | P-DESC | PIC | X(20). | |
| 086 | 03 | FILLER | PIC | X(09) | VALUE SPACES. |
| 087 | 03 | P-QNTY | PIC | ZZZZ9. | |
| 088 | 03 | FILLER | PIC | X(09) | VALUE SPACES. |
| 089 | 03 | P-CODE | PIC | Z99. | |
| 090 | 03 | FILLER | PIC | X(10) | VALUE SPACES. |
| 091 | 03 | P-PRCE | PIC | ZZ9.99. | |
| 092 | 03 | FILLER | PIC | X(10) | VALUE SPACES. |
| 093 | 03 | P-MINI | PIC | Z(4)9. | |
| 094 | 03 | FILLER | PIC | X(12) | VALUE SPACES. |
| 095 | 03 | P-COMNT | PIC | X(7). | |
| 096 | 03 | FILLER | PIC | X(10) | VALUE SPACES. |
| 097 | * | | | | |
| 098 | * | | | | |
| 099 | * | | | | |
| 100 | 01 | C-DATE | PIC | 9(6). | |
| 101 | 01 | PAG-CTR | PIC | 999 | VALUE ZERO. |
| 102 | 01 | LIN-CTR | PIC | 99 | VALUE 25. |
| 103 | 77 | FLAG | PIC | 9 | VALUE ZERO. |
| 104 | * | | | | |
| 105 | | PROCEDURE DIVISION. | | | |
| 106 | * | | | | |
| 107 | * | | | | |
| 108 | * | | | | |
| 109 | | SORT | SRTFIL | | |
| 110 | | | ASCENDING KEY SRT-NO | | |
| 111 | | USING | STOCKF | | |
| 112 | | GIVING | OUTSRT. | | |
| 113 | * | | | | |
| 114 | * | | | | |
| 115 | * | | | | |
| 116 | * | | | | |
| 117 | | START-PROCESSING. | | | |
| 118 | | OPEN INPUT | OUTSRT | | |
| 119 | | OUTPUT | PRINTF. | | |
| 120 | | ACCEPT C-DATE | FROM DATE. | | |
| 121 | | MOVE C-DATE | TO P-DATE. | | |
| 122 | | READ | OUTSRT AT END | | |
| 123 | | MOVE 1 | TO FLAG. | | |
| 124 | * | | | | |

```

125 MAIN-LOOP.
126     PERFORM MOVE-WRITE
127         THRU EXT-MOVE-WRITE
128     UNTIL FLAG EQUAL 1.
129     CLOSE OUTSRT PRINTF.
130     STOP RUN.
131 *
132 MOVE-WRITE.
133     IF      OUT-QTY NOT    > TOT-MIN
134     MOVE    'REORDER' TO  P-COMNT
135     ELSE
136     MOVE    SPACES      TO  P-COMNT.
137     MOVE    OUT-NO      TO  P-NO.
138     MOVE    OUT-DESC    TO  P-DESC.
139     MOVE    OUT-QTY     TO  P-QTY.
140     MOVE    OUT-CODE    TO  P-CODE.
141     MOVE    OUT-PRC     TO  P-PRCE.
142     MOVE    OUT-MIN     TO  P-MINI.
143     IF      LIN-CTR     >    24
144     PERFORM NEW-PAGE.
145     WRITE P-REC FROM DATA-LINE.
146     ADD    1           TO  LIN-CTR.
147     READ OUTSRT AT END.
148     MOVE    1         TO  FLAG.
149 EXT-MOVE-WRITE.
150     EXIT.
151 NEW-PAGE.
152     MOVE SPACES TO P-REC.
153     WRITE P-REC AFTER PAGE.
154     MOVE ZERO TO LIN-CTR.
155     ADD 1 TO PAG-CNTR.
156     MOVE PAG-CNTR TO P-PAG.
157     WRITE P-REC FROM HDR1
158         AFTER 1 LINE.
159     WRITE P-REC FROM HDR2
160         AFTER 1 LINE.
161     WRITE P-REC FROM HDR3
162         AFTER 2 LINE.
163 EXTP.
164     EXIT.
165 *
```

| 00155 | * | |
|--------|---------------|--|
| - CARD | ERROR MESSAGE | |
| 15 | IKF1007I-W | ' FILE ' SHOULD NOT BEGIN A-MARGIN. |
| 15 | IKF1004I-E | INVALID WORD FILE . SKIPPING TO NEXT RECOGNIZABLE WORD. |
| 16 | IKF1003I-W | FILE-CONTROL PARAGRAPH NAME MISSING. ASSUMED PRESENT. |
| 57 | IKF2128I-C | VALUE CLAUSE LITERAL TOO LONG. TRUNCATED TO PICTURE SIZE |
| 111 | IKF1161I-E | STOCKF IS NOT A FILE-NAME IN USING CLAUSE. |
| 109 | IKF4155I-E | SYNTAX REQUIRES 1 TO 16 FILE-NAMES IN SORT USING CLAUSE. DISCARDED. |
| 126 | IKF3001I-E | UNTILE NOT DEFINED. DELETING TILL LEGAL ELEMENT FOUND. |
| 133 | IKF3001I-E | TOT-MIN NOT DEFINED. TEST DISCARDED. |
| 139 | IKF3001I-E | P-QTY NOT DEFINED. DISCARDED. |
| 155 | IKF3001I-E | PAG-CNTR NOT DEFINED. SUBSTITUTING TALLY . |
| 154 | IKF3001I-E | PAG-CNTR NOT DEFINED. DISCARDED. |
| 156 | IKF3001I-E | P-PAG NOT DEFINED. |

٢ - في الصفحات التالية برنامج كوبرول و يظهر بعد ترجمته بواسطة مترجم TI 990 وتبدو الأخطاء الناتجة عن عملية الترجمة COMPILATION في صفحات البرنامج تحت الأسطر الحاوية على أخطاء .
والمطلوب كشف هذه الأخطاء وتصحيحها ليصبح البرنامج خالياً من الأخطاء القاعدية .


```

03 SUBS-CODE          PIC 99.
08 CODEE              VALUES ARE 01 THRU 10.
03 SUBS-AREA          PIC 99.
08 AREA              VALUES ARE 01 THRU 20.
03 SUBS-CNTR          PIC 9(7).
03 SUBS-LREAD         PIC 9(7).
03 FILLER             PIC X(5).

* WORKING-STORAGE SECTION.
*
* 01 DEC-CODE          PIC 99.
*                      USAGE IS COMP-1.
* 01 BIN-VAL REDEFINES DEC-CODE PIC XX.
* 01 CHAR              PIC X(9).
* 01 FLAG              PIC 99.
* 01 FLAGNO            PIC 99.
* 01 DUMMY             PIC X.
* 01 ANSR              PIC X.
* 01 OPTIONS           PIC 9.

* PROCEDURE DIVISION.
*
* STR-----
* 01 OPEN I-O SUBSECT.
* 01 MOVE 18 TO DEC-CODE.
* 01 MOVE BIN-VAL TO CHAR.
* 01 DISPLAY
*
* MENU.
* 01 DISPLAY "LINE 3 POSITION 24 ERASE"
* 01 "LINE 4 POSITION 30"
* 01 "LINE 3 POSITION 24"
* 01 "LINE 4 POSITION 24"

```

```

****
71 1) SYNTAX *****
72 LINE 6 POSITION 27
73 "-----"
74 LINE 8 POSITION 27
75 "32"
76 LINE 10 POSITION 27
77 "10"
78 LINE 12 POSITION 27
79 "12"
80 LINE 14 POSITION 20
81 "-----"
82 LINE 16 POSITION 20.

****
83 1) SCAN RESUME *****
84 ACPT-COD. ACCEPT OPTIONS
85 30018 LINE 12 POSITION 70 PROMPT TAB.
86 30022 IF OPTIONS LESS THAN 4 OR
87 IF OPTIONS GREATER THAN 4
88 CLOSE SUBSFL STOP RUN.
89
90 *
91 *
92 *
93 30036 MOVE HIGH VALUE TO SUBS-NO.
94
95 *****
96 1) RESERVED WORD CONFLICT *****
97 2) SCAN RESUME *****
98 30036 IF OPTIONS = 1 PERFORM INSR-RTN THRU EXT-INS
99 IF OPTIONS = 1 UNTIL SUBS-NO = LOW-VALUES
100 IF OPTIONS = 2 PERFORM UPDAT-RTN THRU EXT-UPD
101 IF OPTIONS = 3 PERFORM DELT-RTN THRU EXT-DEL
102 IF OPTIONS = 4 PERFORM INDR-RTN THRU EXT-IND
103 30078 UNTIL SUBS-NO = LOW-VALUES.
104 GO TO MENU.
105
106 *

```


| | | |
|-----|-------|--|
| 144 | 20132 | *INSRT-RTN. |
| 145 | 20132 | PERFORM ACCEPT-NO. |
| 146 | 20132 | IF SUBS-NO = LOW-VALUES GO TO EXT-INS. |
| 147 | 20134 | IF FLAG = ZERO DISPLAY |
| 148 | 2013C | IF FLAG = SET JUMP TO EXT-INS. |
| 149 | | LINE 20 POSITION 10 EXT-INS. |
| 150 | 20152 | ACCEPT DUMMY GO THRU EXT-INS. |
| 151 | | PERFORM DISPLAY-SCREEN |
| 152 | | DISPLAY SUBS-NO LINE 7 POSITION 45. |
| 153 | | *** |
| 154 | | * (الخط 20) في الموضع 10 من السطر 20 |
| 155 | | * (الخط 20) في الموضع 10 من السطر 20 |
| 156 | | * (الخط 20) في الموضع 10 من السطر 20 |
| 157 | | * (الخط 20) في الموضع 10 من السطر 20 |
| 158 | 2015C | AC-NAME |
| 159 | 2015C | ACCEPT SUBS-NAME LINE 8 POSITION 45 PROMPT TAB. |
| 160 | 20166 | IF SUBS-NAME = SPACES OR SUBS-NAME NUMERIC |
| 161 | | GO TO AC-NAME. |
| 162 | 20174 | *AC-ADDRESS |
| 163 | 20174 | ACCEPT SUBS-ADRES LINE 9 POSITION 45 PROMPT TAB. |
| 164 | 2017E | IF SUBS-ADRES = SPACES GO TO AC-ADDRESS. |
| 165 | | |
| 166 | 20188 | *AC-DAY |
| 167 | 20188 | ACCEPT SUBS-DAY LINE 11 POSITION 45 SIZE 2 |
| 168 | | PROMPT TAB. |
| 169 | 20194 | IF NOT DAY1 GO TO AC-DAY. |
| 170 | | |
| 171 | 201A4 | *AC-MONTH |
| 172 | 201A4 | ACCEPT SUBS-MONTH LINE 12 POSITION 45 SIZE 2 |
| 173 | | PROMPT TAB. |
| 174 | 20180 | IF NOT MONTH1 GO TO AC-MONTH. |
| 175 | | |
| 176 | 201C0 | *AC-YEAR |
| 177 | 201C0 | ACCEPT SUBS-YEAR LINE 13 POSITION 45 SIZE 4 |
| 178 | | PROMPT TAB. |
| 179 | 201DC | IF NOT YEARR GO TO AC-YEAR. |
| 180 | | |
| 181 | 201DC | *AC-CODE. |

202DA
202DA
202DE
202FC
2030B
2031A
2031C
2031E

-۳۵۹-

```

289          EXT-AC.
*****
***** 1) SYNTAX *****
***** 1) SCAN RESUME *****
***** 1) DOUBLE DEFINITION *****
***** 1) EXIT *****
*****
290          *
291          *
292          *
293          *
294          *
295          *
296          *
297          *
298          *
299          *
300          *
301          *
302          *
303          *
304          *
305          *
306          *
307          *
308          *
309          *
310          *
311          *
312          *
313          *
314          *
315          *
316          *
317          *
318          *
319          *
320          *
321          *
322          *
323          *
324          *
325          *
326          *
327          *
328          *
329          *
330          *
331          *
332          *
333          *
334          *
335          *
336          *
337          *
338          *
339          *
340          *
341          *
342          *
343          *
344          *
345          *
346          *
347          *
348          *
349          *
350          *
351          *
352          *
353          *
354          *
355          *
356          *
357          *
358          *
359          *
360          *
361          *
362          *
363          *
364          *
365          *
366          *
367          *
368          *
369          *
370          *
371          *
372          *
373          *
374          *
375          *
376          *
377          *
378          *
379          *
380          *
381          *
382          *
383          *
384          *
385          *
386          *
387          *
388          *
389          *
390          *
391          *
392          *
393          *
394          *
395          *
396          *
397          *
398          *
399          *
400          *
401          *
402          *
403          *
404          *
405          *
406          *
407          *
408          *
409          *
410          *
411          *
412          *
413          *
414          *
415          *
416          *
417          *
418          *
419          *
420          *
421          *
422          *
423          *
424          *
425          *
426          *
427          *
428          *
429          *
430          *
431          *
432          *
433          *
434          *
435          *
436          *
437          *
438          *
439          *
440          *
441          *
442          *
443          *
444          *
445          *
446          *
447          *
448          *
449          *
450          *
451          *
452          *
453          *
454          *
455          *
456          *
457          *
458          *
459          *
460          *
461          *
462          *
463          *
464          *
465          *
466          *
467          *
468          *
469          *
470          *
471          *
472          *
473          *
474          *
475          *
476          *
477          *
478          *
479          *
480          *
481          *
482          *
483          *
484          *
485          *
486          *
487          *
488          *
489          *
490          *
491          *
492          *
493          *
494          *
495          *
496          *
497          *
498          *
499          *
500          *
501          *
502          *
503          *
504          *
505          *
506          *
507          *
508          *
509          *
510          *
511          *
512          *
513          *
514          *
515          *
516          *
517          *
518          *
519          *
520          *
521          *
522          *
523          *
524          *
525          *
526          *
527          *
528          *
529          *
530          *
531          *
532          *
533          *
534          *
535          *
536          *
537          *
538          *
539          *
540          *
541          *
542          *
543          *
544          *
545          *
546          *
547          *
548          *
549          *
550          *
551          *
552          *
553          *
554          *
555          *
556          *
557          *
558          *
559          *
560          *
561          *
562          *
563          *
564          *
565          *
566          *
567          *
568          *
569          *
570          *
571          *
572          *
573          *
574          *
575          *
576          *
577          *
578          *
579          *
580          *
581          *
582          *
583          *
584          *
585          *
586          *
587          *
588          *
589          *
590          *
591          *
592          *
593          *
594          *
595          *
596          *
597          *
598          *
599          *
600          *
601          *
602          *
603          *
604          *
605          *
606          *
607          *
608          *
609          *
610          *
611          *
612          *
613          *
614          *
615          *
616          *
617          *
618          *
619          *
620          *
621          *
622          *
623          *
624          *
625          *
626          *
627          *
628          *
629          *
630          *
631          *
632          *
633          *
634          *
635          *
636          *
637          *
638          *
639          *
640          *
641          *
642          *
643          *
644          *
645          *
646          *
647          *
648          *
649          *
650          *
651          *
652          *
653          *
654          *
655          *
656          *
657          *
658          *
659          *
660          *
661          *
662          *
663          *
664          *
665          *
666          *
667          *
668          *
669          *
670          *
671          *
672          *
673          *
674          *
675          *
676          *
677          *
678          *
679          *
680          *
681          *
682          *
683          *
684          *
685          *
686          *
687          *
688          *
689          *
690          *
691          *
692          *
693          *
694          *
695          *
696          *
697          *
698          *
699          *
700          *
701          *
702          *
703          *
704          *
705          *
706          *
707          *
708          *
709          *
710          *
711          *
712          *
713          *
714          *
715          *
716          *
717          *
718          *
719          *
720          *
721          *
722          *
723          *
724          *
725          *
726          *
727          *
728          *
729          *
730          *
731          *
732          *
733          *
734          *
735          *
736          *
737          *
738          *
739          *
740          *
741          *
742          *
743          *
744          *
745          *
746          *
747          *
748          *
749          *
750          *
751          *
752          *
753          *
754          *
755          *
756          *
757          *
758          *
759          *
760          *
761          *
762          *
763          *
764          *
765          *
766          *
767          *
768          *
769          *
770          *
771          *
772          *
773          *
774          *
775          *
776          *
777          *
778          *
779          *
780          *
781          *
782          *
783          *
784          *
785          *
786          *
787          *
788          *
789          *
790          *
791          *
792          *
793          *
794          *
795          *
796          *
797          *
798          *
799          *
800          *
801          *
802          *
803          *
804          *
805          *
806          *
807          *
808          *
809          *
810          *
811          *
812          *
813          *
814          *
815          *
816          *
817          *
818          *
819          *
820          *
821          *
822          *
823          *
824          *
825          *
826          *
827          *
828          *
829          *
830          *
831          *
832          *
833          *
834          *
835          *
836          *
837          *
838          *
839          *
840          *
841          *
842          *
843          *
844          *
845          *
846          *
847          *
848          *
849          *
850          *
851          *
852          *
853          *
854          *
855          *
856          *
857          *
858          *
859          *
860          *
861          *
862          *
863          *
864          *
865          *
866          *
867          *
868          *
869          *
870          *
871          *
872          *
873          *
874          *
875          *
876          *
877          *
878          *
879          *
880          *
881          *
882          *
883          *
884          *
885          *
886          *
887          *
888          *
889          *
890          *
891          *
892          *
893          *
894          *
895          *
896          *
897          *
898          *
899          *
900          *
901          *
902          *
903          *
904          *
905          *
906          *
907          *
908          *
909          *
910          *
911          *
912          *
913          *
914          *
915          *
916          *
917          *
918          *
919          *
920          *
921          *
922          *
923          *
924          *
925          *
926          *
927          *
928          *
929          *
930          *
931          *
932          *
933          *
934          *
935          *
936          *
937          *
938          *
939          *
940          *
941          *
942          *
943          *
944          *
945          *
946          *
947          *
948          *
949          *
950          *
951          *
952          *
953          *
954          *
955          *
956          *
957          *
958          *
959          *
960          *
961          *
962          *
963          *
964          *
965          *
966          *
967          *
968          *
969          *
970          *
971          *
972          *
973          *
974          *
975          *
976          *
977          *
978          *
979          *
980          *
981          *
982          *
983          *
984          *
985          *
986          *
987          *
988          *
989          *
990          *
991          *
992          *
993          *
994          *
995          *
996          *
997          *
998          *
999          *
1000          *

```

| | | | | | |
|-----------------------------|-------|-------|-------|------------------|----------------|
| 20061 | 2 | NSU | 0 | NUMERIC UNSIGNED | SUBS-AREA |
| 20063 | 7 | NSU | 0 | CONDITION-NAME | AREA |
| 20064 | 7 | NSU | 0 | NUMERIC UNSIGNED | SUBS-CNTR |
| 2007A | 2 | NBS | 0 | NUMERIC UNSIGNED | SUBS-LEAD |
| 2007A | 2 | NBS | 0 | BINARY SIGNED | DEC-CODE |
| 2007A | 2 | ANS | 0 | ALPHANUMERIC | BIN-VAL |
| 2007C | 2 | NSU | 0 | NUMERIC UNSIGNED | CHAR |
| 2007E | 1 | NSU | 0 | NUMERIC UNSIGNED | FLAG |
| 20080 | 2 | NSU | 0 | NUMERIC UNSIGNED | FLD-NO |
| 20082 | 2 | ANS | 0 | ALPHANUMERIC | DUPPLY |
| 20084 | 1 | ANS | 0 | ALPHANUMERIC | ANSR |
| 20086 | 1 | NSU | 0 | NUMERIC UNSIGNED | OPTIONS |
| ILLEGAL PERFORM | ***** | ***** | ***** | ***** | UPDATE-RTN |
| ILLEGAL PERFORM | ***** | ***** | ***** | ***** | THRU EXT-UPD |
| ILLEGAL PERFORM | ***** | ***** | ***** | ***** | AC-ADRES |
| ILLEGAL PERFORM | ***** | ***** | ***** | ***** | AC-LEAD |
| ILLEGAL PERFORM | ***** | ***** | ***** | ***** | DISPLAY-SCREEN |
| READ ONLY BYTE SIZE = | | | | | THRU EXT-SQ |
| READ ONLY BYTE SIZE = | | | | | 20940 |
| READ/WRITE BYTE SIZE = | | | | | 20130 |
| OVERLAY SEGMENT BYTE SIZE = | | | | | 20000 |
| TOTAL BYTE SIZE = | | | | | 20A70 |
| 12 ERRORS | | | | | |
| 2 WARNINGS | | | | | |

القسم الثالث

استخدام الحاسب في كتابة وترجمة وتنفيذ برنامج كويبول

- استخدام منقح النصوص

الفصل الحادي عشر

ترجمة البرنامج وتنفيذه

الفصل الثاني عشر

الفصل الحادي عشر

استخدام منقح النصوص TEXT EDITOR

١١ - ١ ملحة عامة :

يعتبر منقح النصوص TEXT EDITOR من البرامج الأساسية الموجودة في نظام تشغيل أي حاسب آلي يستخدم لغة كوبول .

و يستخدم منقح النصوص كوسيلة لادخال البرامج - الملفات (٥) - إلى الحاسب، أو تعديل برامج - ملفات - قديمة .

إن تقدم التقنية الخاصة بأجهزة الحاسب HARDWARE مع تقدم أنظمة التشغيل والبرامج الأساسية SOFTWARE جعلت استخدام منقح النصوص عبر النمايات - الشاشات - TERMINALS اسهل تقريباً ، وذلك بتقديمها مجموعة أوامر وقواعد ساعدت كثيراً على عملية تحديث البرامج - الملفات - وسنقدم في هذا الفصل عرضاً لاستخدام برنامج منقح النصوص وفق نظامي تشغيل مختلفين . الأول على جهاز IBM 3033S ذي نظام تشغيل يسمى MVS والثاني على جهاز TEXAS INSTRUMENTS ذي نظام التشغيل DX10 . وسنقدم بالبداية لمحة سريعة عن كيفية تشغيل النمايات للوصول إلى برنامج منقح النصوص مع ذكر العمليات الأولية الواجب اتخاذها عند استخدام جهاز IBM للمرة الأولى .

(٥) توجد قيود على طول السجل في الملف تختلف باختلاف الجهاز أو النظام المستخدم، فمثلاً IBM MVS يصل طول ٢٥٥ بايت أما T1990 فيصل إلى ٨٠ بايت .

١١ - ٢ استخدام نظام الحاسب IBM 3033

١١-٢-١ العمليات الواجب اتباعها عند العمل على جهاز IBM 3033 للمرة الأولى :

هناك عدة اجراءات يجب اتخاذها قبل تنفيذ أي برنامج، هذه العمليات أو الاجراءات تتضمن إعطاء قيم معينة يستخدمها الحاسب كل مرة ونحدها نحن فقط مرة واحدة، وتتناول هذه العمليات تحديد نوع الشاشة المستخدمة والطابعة المستخدمة، والاسم الخاص بالمستخدم USER NAME وغيرها من العمليات.

وقبل البدء بتحديد هذه المعطيات نشير إلى أنه هذه المعطيات تختلف من موقع SITE إلى آخر، وسنتحدث عن القيم المعطاة في مركز الحاسب الآلي لمعهد الادارة العامة.

- (أ) عند فتح الشاشة وطلب برنامج SPF تأتي الشاشة رقم ١١-١ نطلب الخيار 0 صفر فتظهر لنا الشاشة رقم ١١-٢ وفيها ثلاثة خيارات OPTIONS 3 ، نطلب الخيار رقم ١ وهو خاص بتوصيف الشاشة نفسها، فتظهر لنا الشاشة رقم ١١-٣ ونطبق القيم المختلفة فيها مع محتوى الشاشة رقم ١١-٣ المرفقة. ثم نضغط مفتاح PF15 للعودة للشاشة رقم ١١-١
- نطلب الخيار 2 في الشاشة رقم ١١-٢ وهو خاص بتحديد قيم وحدات الطباعة المستخدمة، فتظهر لنا الشاشة رقم ١١-٤ نطبق محتوى الشاشة رقم ١١-٤ مع ما يظهر على الشاشة ونعد له مع ملاحظة ما يلي :-
- ١ - يكتب رقم التعريف USER-ID مكان الأحرف XXXX
- ٢ - يكتب اسم التعريف مكان الأحرف YYY
- ٣ - يكتب الاسم مكان الأحرف 'NNNNN'

- ٤ - ونضع الحرف (H) عند التعبير = CLASS
٥ - يوضع USER-ID كاملاً عند التعبير = NOTIFY

نكرر هذه الملاحظات بالنسبة للشاشة رقم ١١ - ٥.
بعد ادخال هذه المعايير نضغط مفتاح PF15 للاتهاء والعودة للشاشة رقم ١١ - ٢.
يمكن طلب الخيار رقم ١١ - ٣ في الشاشة رقم ١١ - ٢ للاطلاع على مفاتيح التحكم. اضغط مفتاح PF15 - أكثر من مرة إذا لزم الأمر - للعودة للشاشة الأساسية رقم ١١ - ١.

(ب) تحديد شكل ومكان طباعة المخرجات OUTPUT LIST
قد نحتاج لطباعة برنامج SOURCE PROGRAM أو ملف معين ، لذلك نطلب الخيار رقم 3.6 اعتباراً من الشاشة الأساسية رقم ١١ - ١ فنظهر لنا الشاشة رقم ١١ - ٦ . نكرر الملاحظات السابقة في تعبئة قيم لغة التحكم JCL الواردة في الفقرة أ نضغط مفتاح PF15 للعودة للشاشة الأساسية رقم ١١ - ١.

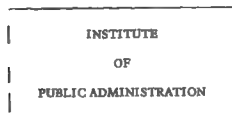
(٥) يختلف باختلاف مركز الحاسب .

SPF-MVS PRIMARY OPTION MENU

SELECTION OPTION ==>

- | | | |
|---|------------|---|
| 0 | SPF PARMS | — SPECIFY TERMINAL AND SPF PARAMENTERS |
| 1 | BROWSE | — DISPLAY SOURCE DATA OR OUTPUT LISTINGS |
| 2 | EDIT | — CREATE OR CHANGE SOURCE DATA |
| 3 | UTILITIES | — PERFORM SPF UTILITY FUNCTIONS |
| 5 | BACKGROUND | — COMPILE, ASSEMBLE, OR LINK EDIT |
| 6 | COMMAND | — ENTER TSO COMMAND OR CLIST |
| T | TUTORIAL | — DISPLAY INFORMATION ABOUT SPF |
| X | EXIT | — TERMINATE SPF USING LIST / LOG DEFAULTS |

PRESS END KEY TO TERMINATE SPF



شکل رقم (۱۱-۱)

SPF PARAMETER OPTION

SELECTION OPTIONS ==>

- 1 TERMINAL — SPECIFY TERMINAL CHARACTERISTICS
- 2 LOG/LIST — SPECIFY SPF LOG AND LIST DEFAULTS
- 3 PF KEYS — SPECIFY PF KEYS FOR 3278BL TERMINAL WITH
 24 PF KEYS

شکل رقم (۱۱-۲)

TERMINAL CHARACTERISTICS

ENTER/VERIFY PARAMETERS BELOW :

| | | | |
|--------------------------------------|-----|-------------|---|
| TERMINAL TYPE | ==> | 3278BL — | (3277 — 3277 / 3275 DISPLAY) (3278 — 3278 / 3276 DISPLAY) (3278BL — 3278 / 3276 DISPLAY BILING- UAL ARABIC) |
| NUMBER OF PF KEYS | ==> | 24 | (12 — TWELVE PROGRAM FUNCTION KEYS) (24 — TWENTY FOUR PROGRAM FUNC- TION KEYS) |
| INPUT FIELD PAD | ==> | N | (N — NULLS) (B — BLANKS) |
| SCREEN FORMAT (3278 MODEL 5 ONLY) | ==> | DATA | (DATA — FORMAT BASED ON DATA WIDTH) (STD — ALWAYS FORMAT 24 LINES BY 80 CHARS) (MAX — ALWAYS FORMAT 27 LINES BY 132 CHARS) |

شكل رقم (١١-٣)

LOG AND LIST DEFAULTS

ENTER / VERIFY PARAMETERS BELOW :

LOG DATASET DEFAULT OPTIONS

```
PROCESS OPTION  ==> D
SYSOUT CLASS    ==> R
LOCAL PRINTER ID ==>
LINES PER PAGE  ==> 60
PRIMARY PAGES    ==> 0
SECONDARY PAGES ==> 0
```

LIST DATASET DEFAULT OPTIONS

```
PROCESS OPTION  ==> J
SYSOUT CLASS    ==> M
LOCAL PRINTER ID ==> LP54E
LINES PER PAGE  ==> 60
PRIMARY PAGES    ==> 100 **
SECONDARY PAGES ==> 200 **
```

(**) EFFECTIVE NEXT SESSION

VALID PROCESS OPTIONS :

J — SUBMIT JOB TO PRINT (AND DELETE)

L — ROUTE TO LOCAL PRINTER (AND DELETE)

D — DELETE DATASET

(DO NOT PRINT)

JOB STATEMENT INFORMATION : (IF OPTION "J" SELECTED)

```
==> / / STDxxxx JOB (STD, xxxx), 'NNNN', CLASS = H, MSGCLASS = Y,
```

```
==> / / NOTIFY = STD xxxx
```

```
==> / / *
```

```
==> / / *
```

شکل رقم (۱۱-۴)

-۳۷۱-

SPECIFY DISPOSITION OF LIST DATASET _____

ENTER / VERIFY PARAMETERS BELOW :

NO LOG DATASET THIS SESSION

LIST OPTIONS FOR THIS SESSION

PROCESS OPTION === J

SYSOUT CLASS === M

LOCAL PRINTER ID === LP54E

VALID PROCESS OPTIONS :

J _ SUBMIT JOB TO PRINT (AND DELETE)

K _ KEEP DATASET (DO NOT
PRINT)

L _ ROUTE TO LOCAL PRINTER (AND DE
LETE)

D _ DELETE DATASET (DO
NOT PRINT)

PRESS ENTER TO COMPLETE SPF TERMINATION

PRESS END KEY TO RETURN TO PRIMARY OPTION MENU

JOB STATEMENT INFORMATION : (IF OPTION "J" SELECTED)

==> // STDxxxxJOB(STD,xxxx), 'NNNN', CLASS = H, MSGCLASS=Y,

==> // NOTIFY=STDxxxx

==> // *

==> // *

شکل رقم (۱۱-۵)

HARDCOPY UTILITY

SELECTION OPTION ==> L

J _ GENERATE JCL TO PRINT OR PUNCH DATASET

L _ ROUTE DATASET TO LOCAL PRINTER

DATASET NAME ==>

| | | | |
|------------------|-----|------|-------------------------|
| DISPOSITION | ==> | KEEP | (KEEP OR DELETE) |
| VOLUME SERIAL | ==> | | (IF NOT CATALOGED) |
| DATASET PASSWORD | ==> | | (IF PASSWORD PROTECTED) |

SYSOUT CLASS ==> M (IF OPTION "J" SELECTED)

LOCAL PRINTER ID ==> LP54E (IF OPTION "L" SELECTED)

JOB STATEMENT INFORMATION : (IF OPTION "J" SELECTED, VERIFY BEFORE
PROCEEDING)

==> // STDxxxxJOB(STD,xxxx), 'NANN' CLASS =H,MSGCLAS=Y,

==> // MSGLEVEL = (2,1)

==> // *

==> // *

شکل رقم (۱۱-۶)

١١-٢-٢ تشغيل النهائيةيات من نوع IBM 3278

يتم تشغيل هذا النوع من الشاشات أو النهائيةيات بالشكل التالي :

- نضع المفتاح الأحمر الموجود على يسار الشاشة على الوضع I
- ننتظر قليلاً حتى تظهر أمامنا الشاشة والضوء الأحمر
- نضغط مفتاح ALT مع مفتاح ATTN بنفس الوقت .
- نكتب بعدها عبارة TSO أي TIME SHARING OPTION
- ننتظر قليلاً وتظهر لنا الرسالة التالية :—

— * ENTER USERID —

- أي أدخل عبارة التعريف الخاصة بنا للحاسب ، نكتب عندها عبارة التعريف الخاصة بنا ولتكن STD4221 مثلاً وننتظر قليلاً .
- تظهر لدينا عدة رسائل لا تعيننا بشيء غالباً .
- ننتظر حتى تظهر لنا عبارة .

READY

- عندها يكون الحاسب مستعداً لتلقي الأوامر منا .
 - نكتب بعدها عبارة SPF ونضغط مفتاح ENTER
 - وتعني STRUCTURE PROGRAM FACILITIES
 - ننتظر قليلاً وتظهر لنا ثلاث نجوم *** نضغط مفتاح ENTER .
 - ننتظر قليلاً لتظهر لنا الشاشة الرئيسية في هذا البرنامج أنظر الشكل المرفق
- رقم ١١-١ .

انطلاقاً من هذه الشاشة باستطاعتنا كتابة برنامج وطباعته والتعديل عليه واستغلال كثير من التسهيلات التي يقدمها الحاسب .

(٥) من الممكن أن يكون هناك كلمة سر PASSWORD فندخل عند ظهور الرسالة المناسبة .

فبعند كتابة أي رقم في الخانة المخصصة لذلك وهي
تأتيينا شاشة ثانية ومنها شاشة فرعية ثالثة... الخ.
SELECT OPTION

إغلاق النهاثيات - الشاشات من نوع IBM3278

- أينما كنا نضغط مفتاح انتهاء PF15 للوصول إلى الشاشة رقم ١١-١.
- بعد الوصول للشاشة رقم ١١-١ نكتب حرف x في مكان الاختيار
ENTER ونضغط
READY ننتظر قليلاً لتظهر لنا عبارة
LOGOFF ونضغط
ENTER نكتب تعليمة
ننتظر قليلاً بعدها نطفىء مفتاح التشغيل الأحمر، أي نرجعه للوضع 0
لا تنسى أوراقك الخاصة قرب الشاشات لأنها قد تعتبر أوراقاً
عديمة القيمة وتتلف.
حافظ على نظافة المكان قبل مغادرتك صالة الحاسب.

مجموعة أزرار التحكم FUNCTION KEYS

- يوجد على يمين لوحة الأزرار KEY BOARD مجموعة من المفاتيح معنونة كالتالي :
- PF13 , PF14 , , PF24 وكذلك أعلى مفاتيح الأرقام مجموعة من أزرار التحكم
PF1 , PF2 , PF3 ... تستخدم مع مفتاح ALT
ما يهمنا منها هي مجموعة أزرار التحكم من 13 إلى 24 لأن مفاتيح التحكم من
(١) وحتى (١٢) مطابقة في عملها للمفاتيح من 13 إلى 24 انظر للجدول التالي لتلاحظ
عمل هذه المفاتيح: الشكل رقم ١١-٧.

PROGRAM FUNCTION KEY DEFINITION

ENTER / VERIFY PARAMETERS BELOW FOR 3278T TERMINAL WITH 24 PF KEYS :

| | | | |
|----------|--------|---|--------|
| PF1 ==> | HELP | المساعدة PF13 ==> | HELP |
| PF2 ==> | SPLIT | الفصل الشاشة إلى اثنين PF14 ==> | SPLIT |
| PF3 ==> | END | لانتهاء عمل PF15 ==> | END |
| PF4 ==> | RETURN | للمودة للشاشة الأساسية PF16 ==> | RETURN |
| PF5 ==> | FIND | لايجاد حرف أو مجموعة حروف PF17 ==> | FIND |
| PF6 ==> | CHANGE | لتغيير حرف أو مجموعة حروف PF18 ==> | CHANGE |
| PF7 ==> | UP | لارجاع ما عرض على الشاشة للخلف PF19 ==> | UP |
| PF8 ==> | DOWN | لتقديم الشاشة للأمام PF20 ==> | DOWN |
| PF9 ==> | SWAP | للاتصال من شاشة للثانية PF21 ==> | SWAP |
| PF10 ==> | LEFT | لتحرك محتوى الشاشة لليسار PF22 ==> | LEFT |
| PF11 ==> | RIGHT | لتحرك محتوى الشاشة لليمين PF23 ==> | RIGHT |
| PF12 ==> | CURSOR | PF24 ==> | CURSOR |

VALID OPTIONS : (BLANK RESETS TO DEFAULT OPTION)

HELP SPLIT END RETURN FIND CHANGE
UP DOWN SWAP LEFT RIGHT CURSOR
NOP PRINT PRINT_HI

OR EQUATE TO EDIT / BROWSE COMMAND :

>CMD ... (PRIMARY COMMAND) EXAMPLE: PF10 ==>> TABS ON

: CMD ... (EDIT LINE COMMAND) EXAMPLE: PF11 ==> : I

شكل رقم (١١-٧)

١١-٣ استخدام منقح النصوص TEXT EDITOR

من خلال شاشات IBM 3278

١١-٣-١ كتابة برنامج جديد :

انطلاقاً من الشاشة الرئيسية رقم ١١-٨ نكتب الخيار 2 وهو EDIT، هذا الخيار يتيح لنا كتابة برنامج جديد بلغة الكوبول أو تعديل برنامج سبق كتابته فعند إدخال ٢ والضغط على مفتاح ENTER تظهر لدينا الشاشة رقم ١١-٩.

ونتأكد من أنه حقل PROJECT يحتوي على STDxxxx

وأن حقل LIBRARY يحتوي على TEST

وأن حقل TYPE يحتوي على COBOL

بعد التأكد نكتب اسم البرنامج الجديد بحقل MEMBER ولنفترض أنه PROL
نضغط مفتاح ENTER فيفتح ملف جديد باسم PROL ضمن المكان المخصص للكتابة
DATASET انظر الشاشة رقم ١١-١٠.

نضغط على مفتاح NEW LINE لنصل للسطر الأول ونكتب عبارة COLS لتحديد
المواقع التي نرغب بالعمل عليها. فنحن عادة في الكوبول نبتدىء بالعمود الثامن كما
مر معنا في الفصل الأول. نضغط ENTER لنحصل على الشاشة رقم ١١-١١ نبدأ
بالعمود رقم ٨ بكتابة البرنامج المعطى لنا كما هو واضح في الشاشات رقم ١١-٥،
١١-٦، ١١-٧، ١١-٨.

يلاحظ من الأشكال أن سطر نهاية الملف BOTTOM OF DATA يظهر عقب
السطر الأول وللتغلب على ذلك نرجع لأعلى الشاشة بضغط مفتاح PF24 بعدها نضغط

(٥) هذه الأسماء تبدل من موقع لآخر، لذلك راجع الملحق بشأن إعطاء الاسم المناسب؛

مفتاح NEW LINE لنصل للسطر الذي ترغب بادخال أسطر جديدة بعده نكتب الأمر I أي INSERT و يعني ادخل أسطرا بعد هذا السطر فاذا كتبنا I20 — أي أدخل أو افتح ٢٠ سطراً جليداً — نضغط ENTER لنحصل على صفحة كاملة لادخال البرنامج انظر الشاشة رقم ١١-١٤ والسهم الدال على مكان كتابة التعليمه I20 وكذلك الشكل رقم ١١-١٥ .

عند الانتهاء من كتابة البرنامج نضغط مفتاح PF15 لحفظ برنامج تحت اسم PROL مثلاً نضغط PF15 مرة ثانية لنعود إلى الشاشة الرئيسية رقم ١١-٨ .

SPF-MVS PRIMARY OPTION MENU

SELECTION OPTION ==>

- 0 SPF PARMS — SPECIFY TERMINAL AND SPF PARAMENTERS
- 1 BROWSE — DISPLAY SOURCE DATA OR OUTPUT LISTINGS
- 2 EDIT — CREATE OR CHANGE SOURCE DATA
- 3 UTILITIES — PERFORM SPF UTILITY FUNCTIONS
- 4 FOREGROUND — COMPILE, ASSEMBLE, LINK EDIT, OR DEBUG
- 5 BACKGROUND — COMPILE, ASSEMBLE, OR LINK EDIT
- 6 COMMAND — ENTER TSO COMMAND OR CLIST
- 7 SUPPORT — TEST DIALOG OR CONVERT MENU / MESSAGE FORMATS
- T TUTORIAL — DISPLAY INFORMATION ABOUT SPF
- X EXIT — TERMINATE SPF USING LIST / LOG DEFAULTS

PRESS END KEY TO TERMINATE SPF

| |
|-----------------------|
| INSTITUTE |
| OF |
| PUBLIC ADMINISTRATION |

شکل رقم (۱۱-۸)

EDIT - ENTRY PANEL

ENTER / VERIFY PARAMETERS BELOW :

SPF LIBRARY :

PROJECT ==> STD xxxx

LIBRARY ==> TEST ==> ==> ==>

TYPE ==> COBOL

MEMBER ==> (BLANK FOR MEMBER SELECTION LIST)

OTHER PARTITIONED OR SEQUENTIAL DATASET :

DATASET NAME ==>

VOLUME SERIAL ==> (IF NOT CATALOGED)

DATSET PASSWORD ==> (IF PASSWORD PROTECTED)

PROFILE NAME ==> (BLANK DEFAULTS TO DATASET TYPE)

شکل رقم (۹-۱۱)

-۳۸۰-

EDIT _PAY. TEST. COBOL (PRG1)

COLUMNS 001 072

COMMAND INPUT ==>

SCROLL ==> HALF

***** TOP OF DATA **

***** BOTTOM OF DATA *****

شکل رقم (۱۱-۱۰)

```

EDIT _ PAY. TEST. COBOL (PRG1)                                COLI
COMMAND INPUT ==                                                SCROLL ==
***** ***** TOP OF DATA *****
= COLS  _+_1+_2+_3+_4+_5+_6+_
. . . . .
***** ***** BOTTOM OF DATA *****

```

شکل رقم (۱۱-۱۱)


```

EDIT _ PAY. TEST. COBOL (PRG1)                                COLI
COMMAND INPUT ==                                                SCROLL =

*****      ***** TOP OF DATA *****
= COLS      _+_1+_2+_3+_4+_5+_6+_
000100      IDENTIFICATION DIVISION.
*****      ***** BOTTOM OF DATA *****

```

شکل رقم (۱۱-۱۳)

```

EDIT _ PAY. TEST. COBOL (PRG1)                                SCROLL =
COMMAND INPUT ==                                                SCROLL =

*****      ***** TOP OF DATA *****
= COLS      _+_1+_2+_3+_4+_5+_6+_
I20100      IDENTIFICATION DIVISION.
*****      ***** BOTTOM OF DATA *****

```

شکل رقم (۱۱-۱۴)

EDIT _ PAY. TEST. COBOL (PRG1)

COLI

COMMAND INPUT ==

SCROLL ==

***** TOP OF DATA *****

= COLS _ + _ 1 _ + _ 2 _ + _ 3 _ + _ 4 _ + _ 5 _ + _ 6 _ + _

000100

IDENTIFICATION DIVISION.

يظهر الأسطر الجديدة بعد تنفيذ أمر I20

شكل رقم (١١-١٥)

١١-٣-٢ تنقيح برنامج - ملف - موجود سابقاً *

إن إدخال البرنامج إلى الحاسب لا يعتبر نهاية العمل بالنسبة لهذا البرنامج فهو قد يحتوي على أخطاء قاعدية SYNTAX ERROR ناتجة إما عن خطأ في إدخال عبارات البرنامج؛ أو عن صياغة هذه العبارات أصلاً. مهماً يكن من أمر فإن متطلبات صيانة البرامج تفرض تعديل هذا البرنامج ولو كان خالياً من الأخطاء القاعدية.

ومن أجل تعديل برنامج معين في نظام IBM3033 باستخدام نهائيات من نوع 3278 نتبع مايلي :

اعتباراً من الشاشة الأساسية رقم ١١-١ نطلب الخيار رقم ٢ وهو برنامج EDIT فتأتي الشاشة رقم ١١-١٦ نكتب اسم البرنامج الذي نريد التعديل عليه ، نضغط مفتاح ENTER فتأتي صفحة جديدة تحتوي على البرنامج المراد العمل عليه .

إن عمليات التعديل قد تتناول سطراً أو أسطراً معينة بالبرنامج أو قد تتناول عبارة محددة متكررة في البرنامج ، أو تعالج البرنامج ككل لذلك سنقسم شرح طريقة العمل على منقح النصوص TEXT EDITOR إلى الأقسام التالية . :

أ - الأوامر التي تعالج الأسطر LINE COMMANDS

ب - أوامر أساسية عامة GENERAL COMMANDS

ج - أوامر البحث عن حرف أو سلسلة أحرف ، أوامر تبديل حرف أو سلسلة أحرف FINDING AND CHANGING COMMANDS

د - أوامر متقدمة .

(٥) يحتوي برنامج EDIT العديد من الأوامر الإضافية المتقدمة لذا من أجل زيادة الاطلاع يرجع إلى IBM MANUALS .

(٥٥) Time Sharing Option 3270 Display Support and Structured Programming Facility Version 202

: Program Reference Manual.

EDIT — ENTRY PANEL

ENTER / VERIFY PARAMETERS BELOW :

SPF LIBRARY :

PROJECT ==> STD4253
LIBRARY ==> TEST ==> ==> ==>
TYPE ==> COBOL
MEMBER ==> (BLANK FOR MEMBER SELECTION LIST)
—

OTHER PARTITIONED OR SEQUENTIAL DATASET :

DATASET NAME ==>
VOLUME SERIAL ==> (IF NOT CATALOGED)

DATASET PASSWORD ==> (IF PASSWORD PROTECTED)

PROFILE NAME ==> (BLANK DEFAULTS TO DATASET TYPE)

شكل رقم (١١ - ١٦)

والجدول التالي يلخص هذه الأوامر :

| | أوامر سطرية LINE COMMANDS | أوامر أساسية Primary Commands |
|--|---|--|
| Basic Line Commands أوامر الأسطر الأساسية | COLS, M-MOVE I-INSERT, C-COPY D-DELETE, A-AFTER R-REPEAT, B-BEFORE | |
| General Command أوامر عامة | | Locate Print Number Reset RENUM Submit UNNUM Save Autonum Cancel |
| Find, Change أمر البحث والتبديل | | FIND CHANGE |
| أوامر متقدمة Advanced Features | (— COLS LEFT)— COLS RIGHT | Copy, Profile Move HEX Create Replace |

قبل البدء في استعراض هذه الأوامر نذكر أنه يمكن تحريك إشارة التوضيح —
الدليل CURSOR إلى أي مكان على الشاشة باستخدام أسهم التحريك الأربعة
الموجودة على لوحة المفاتيح KEYBOARD

(أ) الأوامر الأساسية التي تعالج أسطر BASIC LINE COMMANDS

هذه الأوامر هي :

تحديد المواقع — الأعمدة COLS (COLUMNS)

إدخال سطر أو أسطر جديدة (INSERT) I

حذف سطر أو أكثر (DELETE) D

تكرار سطر (REPEAT) R

نقل سطر من مكان لآخر (MOVE) M

نسخ سطر (COPY) C

بعد (AFTER) A

قبل (BEFORE) B

جميع هذه الأوامر تكتب اعتباراً من الموقع POSITION الأول من السطر — أو الأسطر — المراد العمل عليها وسنسميها منطقة الأوامر السطرية . وتكتب الأوامر فوق الأرقام الموجودة أصلاً في السطر.

١ — أمر COLS

هذا الأمر يحدد الأعمدة أو المواقع (POSITIONS) للسطر، ويفيد هذا الأمر لمعرفة الموقع الذي يجب أن تبدأ فيه كتابة السطر، ففي برنامج كوبول يجب أن تبدأ الفقرات وعناوين الأقسام DIVISIONS و SECTIONS في الهامش أ MARGIN A أي من الموقع الثامن إلى الحادي عشر . بينما التعليمات STATEMENTS في الهامش ب MARGIN B أي من العمود ١٢ إلى ٧٢ . وتوضح النجمة (•) في العمود السابع للدلالة على أن هذا السطر مخصص للملاحظات .

ويمكن كتابة السطر الأول باستخدام أمر COLS بعدها تكتب باقي الأسطر بالقياس . انظر الشكل رقم (١١-١٧) .

EDIT _ STD4253. TEST. COBOL (PROGA) _ 01.20

COLUMNS 001 072

COMMAND INPUT ==>

SCROLL ==> HALF

***** TOP OF DATA *****

000100 IDENTIFICATION DIVISION

= COLS _ + _ 1 _ + _ 2 _ + _ 3 _ + _ 4 _ + _ 5 _ + _ 6 _ + _ 7 _

000200 PROGRAM-ID. PROGA.

000300 * THIS PROGRAM PRINTS IPA_

000400 * NAME AND ADDRESS

000500 *

000600 ENVIRONMENT DIVISION.

000700 CONFIGURATION SECTION.

000800 SOURCE-COMPUTER. IBM-370.

000900 OBJECT-COMPUTER. IBM-370.

001000 *

001100 INPUT-OUTPUT SECTION.

001200 FILE-CONTROL.

001300 SELECT CARD-F ASSIGN TO UT-S-CARD.

001400 SELECT PRINT-F ASSIGN TO UT-PRFL.

001500 *

001600 DATA DIVISION.

001700 FILE SECTION.

001800 FD CARD-F RECORD CONTAINS 80 CHARACTERS

001900 LABEL RECORD IS OMITTED.

002000 01 CARD-REC PIC X (80).

شکل رقم (۱۱-۱۷)

٢- أمر INSERT إدخال سطر (I)

يستخدم هذا الأمر لإدخال سطر أو أكثر إما في نهاية البرنامج - الملف - أو بين سطرين معينين ضمن البرنامج.

ويكتب الحرف In في السطر الذي يراد إضافة أسطر جديدة بعده ثم يضغط مفتاح ENTER فتحصل إزاحة بقدر سطر - أو أكثر حسب قيمة n - تتيح إدخال الأسطر المطلوبة. هذه الأسطر الجديدة تعامل كأبي سطر قديم موجود أصلاً وتعطى رقماً متسلسلاً.

بعد إدخال السطر الجديد نضغط ENTER نجد أن فراغاً آخر - إزاحة ثانية - قد تمت تسمح بإدخال سطر جديد آخر، فإذا لم يكن هناك أي أسطر جديدة يمكن ضغط مفتاح ENTER مرة ثانية لانتهاء أمر I المحدث سابقاً.

انظر الشكل رقم (١١-١٨) والشكل رقم (١١-١٩)

٣- أمر حذف سطر DELETE D

قد نحتاج أحياناً لحذف سطر أو أكثر من النص. فإذا أردنا حذف سطر معين ما علينا إلا وضع حرف D في بداية السطر المراد حذفه وضغط مفتاح ENTER ليحذف هذا السطر.

أما عندما نريد حذف مجموعة أسطر متسلسلة فنكتب عبارة DD في بداية الأسطر المراد حذفها وDD في السطر الأخير من المجموعة. وعند ضغط ENTER تحذف مجموعة الأسطر هذه.

كما يمكن الحذف بطريقة أخرى، وذلك بأن نضع الأمر D وأمامه عدد الأسطر المطلوب حذفها Dn وعند ضغط مفتاح ENTER تحذف هذه الأسطر.

انظر الشكل رقم (١١-٢٠)

EDIT — STD4253. TEST. COBOL (PROGA) — 01.20 ————— COLUMNS 001 072

COMMAND INPUT ==>

SCROLL ==> HALF

003100 *

003200 PERFORM READ-WRITE THRU EXT

003300 UNTIL END-OF-JOB = 'YES'.

003400 CLOSE CARD-F PRINT-F.

003500 STOP RUN.

003600 *

003700 READ-WRITE.

003800 READ CARD-F AT END MOVE 'YES' TO

003900 END-OF-JOB GO TO EXT.

004000 *

004100 MOVE CARD-REC TO PRINT-LINE.

004200 WRITE PRINT-LINE AFTER 1 LINE.

004300 *

004400 EXT.

004500 EXIT.

***** ***** BOTTOM OF DATA *****

شکل رقم (۱۱-۱۸)

EDIT — STD4253. TEST. COBOL (PROGA) — 01.20 ————— COLUMNS 001 072

COMMAND INPUT ==>

SCROLL ==> HALF

003100 •

003200 PERFORM READ-WRITE THRU EXT

003300 UNTIL END-OF-JOB = 'YES'.

003400 CLOSE CARD-F PRINT-F.

003500 STOP RUN.

IIIIII

003600 •

003700 READ-WRITE.

003800 READ CARD-F AT END MOVE 'YES' TO

003900 END-OF-JOB GO TO EXT.

004000 •

004100 MOVE CARD-REC TO PRINT-LINE.

004200 WRITE PRINT-LINE AFTER 1 LINE.

004300 •

004400 EXT.

004500 EXIT.

***** ***** BOTTOM OF DATA *****

شکل رقم (۱۱-۱۹)

-۳۹۳-

EDIT — STD4253. TEST. COBOL (PROGA) — 01.20

COMMAND INPUT ==> . SCROLL ==> HALF

```
003100  *
003200      PERFORM READ-WRITE THRU EXT
003300      UNTIL END-OF-JOB = 'YES'.
003400      CLOSE CARD-F PRINT-F.
003500      STOP RUN.

DD1600  *
003700      READ-WRITE.
003800      READ CARD-F AT END MOVE 'YES' TO
003900      END-OF-JOB GO TO EXT.

DD4000  *
004100      MOVE CARD-REC TO PRINT-LINE.
004200      WRITE PRINT-LINE AFTER 1 LINE.
004300  *
004400      EXT.
004500      EXIT.

***** ***** BOTTOM OF DATA *****
```

شکل رقم (۱۱ - ۲۰)

٤ - أمر التكرار : REPEAT R

هذا الأمر يفيد بتكرار سطر معين مرّة أو عدة مرات مباشرة بعد السطر المكتوب عليه هذا الأمر.

فإذا كتبنا R في السطر رقم ٢٠ وبعد ضغط ENTER يتكرر السطر رقم ٢٠ مرّة ثانية؛ ولو أردنا تكرار السطر عدد من المرات نكتب Rn حيث أن n عدد مرات التكرار.

ويمكن أن نكرر مجموعة من الأسطر BLOCK OF LINES عدد من المرات وذلك بكتابة الأمر RR في السطر الأول من المجموعة و RR في السطر الأخير من المجموعة وبعد ضغط ENTER تتكرر هذه المجموعة مرّة ثانية مباشرة بعد المجموعة الأساسية ويمكن أن تتكرر n مرّة RRn .

انظر الشكل رقم (١١-٢١) والشكل رقم (١١-٢٢).

٥ - أمر MOVE : M

هذا الأمر يسمح بنقل سطر أو مجموعة أسطر من مكان معين في البرنامج - الملف - إلى مكان آخر.

ويستخدم بوضع الأمر M في السطر المراد نقله ثم الانتقال إلى المكان الآخر المراد نقل السطر إليه نضع الأمر A أو B عند السطر المناسب ويعني الأمر A أن يتم نقل السطر المميز بالأمر M إلى مكان آخر من البرنامج إلى ما بعد السطر المميز بالأمر A (AFTER)، أما إذا وضعنا الأمر B في سطر معين فهذا يعني أننا نريد نقل السطر المميز بالأمر M لما قبل السطر المميز بالأمر B .

كما ويمكن نقل مجموعة من الأسطر لمكان آخر من البرنامج وذلك بكتابة الأمر MM في السطر الأول من المجموعة أو MM في السطر الأخير من المجموعة مع كتابة B أو A في مكان آخر من البرنامج، وعند ضغط مفتاح ENTER تنتقل مجموعة الأسطر من

مكانها إلى ما بعد A ، أو ما قبل B السطر المميز بالحرف A أو B .
انظر الشكل رقم (١١-٢٣)

٦ - أمر COPY : C

يستخدم هذا الأمر لنسخ سطر أو مجموعة أسطر من مكان معين في البرنامج -
الملف - إلى مكان آخر.

وعمله شبيه جداً بعمل الأمر MOVE إلا أن الأخير COPY يطبع نسخه من السطر
أو الأسطر بمكان آخر من البرنامج بينما MOVE ينقل السطر أو الأسطر من مكان معين
في البرنامج إلى مكان آخر ويستخدم الأمران B أو A بالطريقة السابقة نفسها من أجل
نسخ السطر أو الأسطر قبل أو بعد سطر معين .

كما ويمكن نسخ مجموعة أسطر بالأمر CC كما هو وارد في الأمر MM مثال :
لواردنا نسخ الأسطر من رقم (٣٦١٠) إلى (٣٦٤٠) بمكان آخر بعد السطر
(٣٩٠٠) في البرنامج .

نكتب الأمر CC في السطر ٣٦١٠ وكذلك الأمر CC في السطر (٣٦٤٠) ونذهب
للسطر ٣٩٠٠ ونكتب عليه الأمر A نضغط مفتاح ENTER فتنسخ الأسطر من ٣٦١٠
إلى ٣٦٤٠ بعد السطر رقم ٣٩٠٠ وتأخذ تسلسلاً جديداً وتبقى الأسطر من ٣٦١٠ -
٣٦٤٠ كما هي عليه .

انظر المثال شكل رقم (١١-٢٤)

كما ويمكن نسخ سطر معين عدداً من المرات وذلك بكتابة Cn بحيث تدل n على
عدد المرات المطلوبة لنسخ هذا السطر .

* ملاحظة مهمة :

- ١ - الأمر C والأمر M يجب أن يقرن بالأمر A أو الأمر B .
- ٢ - لا يمكن مع أمر M أو C ورود أكثر من أمر A أو أمر B أو الأمرين A و B معاً .

EDIT — STD4253. TEST. COBOL (PROGA) — 01.20 ————— COLUMNS 001 072

COMMAND INPUT ==> SCROLL ==> HALF

```
003200      PERFORM READ-WRITE THRU EXT
003300      UNTIL END-OF-JOB = 'YES'.
003400      CLOSE CARD-F PRINT-F.
003500      STOP RUN.
003600      *
003700      READ-WRITE.
003800      READ CARD-F AT END MOVE 'YES' TO
003900      END-OF-JOB GO TO EXT.
004000      *
004100      MOVE CARD-REC TO PRINT-LINE.
004200      WRITE PRINT -LINE AFTER 1 LINE.
004300      *
004400      EXT.
004500      EXIT.
***** ***** BOTTOM OF DATA *****
```

شکل رقم (۱۱-۲۱)

EDIT _ STD4253. TEST. COBOL (PROGA) _ 01.20 _ COLUMNS 001 072

COMMAND INPUT ==> SCROLL ==> HALF

003200 PERFORM READ-WRITE THRU EXT

003300 UNTIL END-OF-JOB = 'YES'.

003400 CLOSE CARD-F PRINT-F-

003500 STOP RUN.

003600 *

003700 READ-WRITE.

003800 READ CARD-F AT END MOVE 'YES' TO

003900 END-OF-JOB GO TO EXT.

004000 *

004100 MOVE CARD-REC TO PRINT-LINE.

004110 MOVE CARD-REC TO PRINT-LINE.

004200 WRITE PRINT-LINE AFTER 1LINE.

004300 *

004400 EXT.

004500 EXIT.

***** BOTTOM OF DATA *****

شکل رقم (۱۱-۲۲)

-۳۹۸-

EDIT _ STD4253. TEST. COBOL (PROGA) _ 01.20 _ COLUMNS 001 072

COMMAND INPUT ==>

SCROLL ==> HALF

003200 PERFORM READ-WRITE THRU EXT

003300 UNTIL END-OF-JOB = 'YES'.

003400 CLOSE CARD-F PRINT-F.

003500 STOP RUN.

003600 •

B3700 READ-WRITE.

003800 READ CARD-F AT END MOVE 'YES' TO

003900 END-OF-JOB GO TO EXT.

MM4000 •

004100 MOVE CARD-REC TO PRINT-LINE.

004200 WRITE PRINT-LINE AFTER 1 LINE.

MM4300 •

004400 EXT.

004500 EXIT.

***** ***** BOTTOM OF DATA *****

شکل رقم (۱۱-۲۳)

EDIT — STD4253. TEST. COBOL (PROGA) — 01.20

COMMAND INPUT ==>

SCROLL == HALF

003200 PERFORM READ-WRITE THRU EXT

003300 UNTIL END-OF-JOB = 'YES'.

003400 CLOSE CARD-F PRINT-F.

003500 STOP RUN.

003600 *

CC3610 *

003620 MOVE CARD-REC TO PRINT-LINE.

003630 WRITE PRINT-LINE AFTER 1 LINE.

CC3640 *

003700 READ-WRITE.

003800 READ CARD-F AT END MOVE 'YES' TO

A03900 END-OF-JOB GO TO EXT.

004400 EXT.

004500 EXIT.

***** ***** BOTTOM OF DATA *****

شکل رقم (۱۱-۲۴)

ب — أوامر عامة GENERAL COMMANDS

هذه الأوامر هي أوامر أولية — أساسية — تكتب في مكان إدخال الأوامر في السطر الثاني من شاشة منقح النصوص TEXT EDITOR.

والشكل العام لها : Command [operand] > COMMAND INPUT
وستعرض منها ما يلي :

| | |
|--------|--------|
| LOCATE | RESET |
| NUMBER | SAVE |
| RENUM | SUBMIT |
| UNNUM | CANCEL |

١ — أمر الوصول لسطر معين LOCATING LINES

يستخدم هذا الأمر من أجل الوصول لسطر معين في البرنامج — الملف — والشكل العام له :

COMMAND INPUT > LOCATE line - number

ويمكن اختصار LOCATE إلى LOC أو L

انظر الشكل رقم (١١—٢٥)

٢ — توليد وضبط الأرقام المتسلسلة :

إن منقح النصوص TEXT EDITOR يولد آلياً أرقاماً متسلسلة بالنسبة للأسطر الجديدة المضافة بأمر INSERT أو أمر COPY أو يعاد ترقيم البرنامج كله عندما يحفظ على وسط التخزين الثانوي .

EDIT — STD4253. TEST. COBOL (PROGA) — 01.20 ————— COLUMNS 001 072

COMMAND INPUT ==> L 32

SCROLL ==> HALF

```
003200      PERFORM READ-WRITE THRU EXT
003300      UNTIL END-OF-JOB  = 'YES'.
003400      CLOSE CARD-F PRINT-F.
003500      STOP RUN.
003600      *
003640      *
003700      READ-WRITE.
003800      READ CARD-F AT END MOVE 'YES' TO
003900      END-OF-JOB GO TO EXT.
004000      *
004100      MOVE CARD-REC TO PRINT-LINE.
004200      WRITE PRINT-LINE AFTER 1 LINE.
004300      *
004400      EXT.
004500      EXIT.
***** ***** BOTTOM OF DATA *****
```

شکل رقم (۱۱-۲۵)

صيغة الأرقام المتسلسلة :

تتولد الأرقام المتسلسلة آلياً بالنسبة لبرنامج كوبول — تشكل هذه الأرقام الخانات الستة الأولى من كل سطر كوبول تبدأ عادة بالرقم ١٠٠ وتزداد بمعدل ١٠٠ لكل سطر تالي وعند إضافة أسطر جديدة بأمر INSERT تزداد بمعدل ١٠ لتحافظ على التسلسل المنطقي للبرنامج.

في الواقع هناك تسلسل آخر خاص بمنهج النصوص وهو STANDARD الترقيم القياسي . وهو غالباً ما يطابق ترقيم الكوبول . ويمثل الخانات الست الأولى من سطر منهج النصوص .

ضبط تسلسل الترقيم :

يمكن ضبط تسلسل الترقيم بأمر NUMBER والشكل العام للأمر.

COMMAND INPUT > NUMBER ON/OFF [COBOL] [STD]

• ويمكن اختصار الأمر NUMBER إلى NUMB

والتعبير COBOL إلى COB

أما التعبير STD فيعني STANDARD

عندما يكون الأمر NUMBER على الوضع ON أي في حالة عمل ، فإن هذا الأمر يعمل تحقق Verify لأسطر البرنامج بحيث يتأكد من أن تسلسل الأسطر هو تصاعدي ، وكذلك فإن هذا الأمر إذا وضع على الوضع ON فإنه يعيد ترقيم الأسطر غير المرقمة أو الخارجة عن التسلسل العام ، لكنه لا يعادل تسلسل أرقام الأسطر الموجودة أصلاً .

اما عبارة COBOL فتعني أن الترقيم والتسلسل يتبع قواعد لغة كوبول . أما عبارة STD فتعني أن الترقيم يتبع الترقيم القياسي العام ويمكن جمع التسلسلين معاً .
و يلغى عمل هذا الأمر بكتابة العبارة OFF

> NUMBER OFF

وهناك ثلاثة أوامر تتبع في عملها للأمر NUMBER وهي :

— أمر إعادة الترقيم RENUM

— أمر إلغاء الترقيم UNNUM

— أمر الترقيم الذاتي AUTONUM

ولا تعمل هذه الأوامر إلا إذا كان الأمر NUMBER على الوضع ON

٣ — أمر إعادة الترقيم RENUM

والشكل العام له :

COMMAND INPUT > RENUM [STD] [COBOL]

ويعمل هذا الأمر على إعادة ترقيم الأسطر كلها مباشرة ، و يبدأ الترقيم بالسطر الأول وبالرقم ١٠٠ ويزداد بمعدل ١٠٠ لكل سطر تالي .

٤ — أمر إلغاء الترقيم UNNUM

ويعمل هذا الأمر على إلغاء كل حقول الترقيم و يضع مكانها فراغاً BLANKS والشكل العام له :

COMMAND INPUT > UNNUM

٥ — أمر الترقيم الذاتي AUTONUM

يعمل هذا الأمر على إعادة ترقيم البرنامج — الملف — كل مرة قبل أن يحفظه على وسط التخزين الثانوي ، و يلغى عمله بكتابة التعبير OFF والشكل العام .

COMMAND INPUT > AUTONUM OFF/ON

٦ - أمر الطباعة PRINT

يمكن إرسال البرنامج آلياً لطابعته عند الخروج من البرنامج بفتح مفتاح التحكم PF15 وذلك بوضع أمر الطباعة على وضع العمل ON، والشكل العام للأمر:

COMMAND INPUT > PRINT ON/OFF

لكن يفضل وضعه على الحالة OFF لأنه من المربك أن نطبع كل مرة نطلبه فيها للتعديل.

٧ - أمر RESET

يمكن أن يظهر على البرنامج المطلوب في TEXT EDITOR وسائل تحذير أو إعلام مختلفة أو عرض لأوضاع بعض الأوامر، أو العبارات الناتجة عن استخدام بعض الأوامر مثل أمر COLS أو CHANGE مما يؤدي أحياناً إلى إرباك المبرمج من كثرة الرسائل، ونستخدم أمر RESET حتى نلغي كل الرسائل والعبارات التي لا تتبع لميكال البرنامج والشكل العام للأمر:

> RESET

ويمكن اختصار الأمر RESET بالأحرف RES

٨ - أمر SUBMIT للتنفيذ :

يستخدم هذا الأمر لإرسال النص المكتوب للتنفيذ EXECUTE في المكان المخصص لذلك ويسمى BACKGROUND، والنص المكتوب غالباً ما يحتوي على عبارات لغة التحكم بالعمل JCL. والشكل العام للأمر:

> SUBMIT

ويختصر بالأحرف SUB

٩ - أمر الحفظ SAVE

من المعروف أن مفتاح التحكم PF15 ينهي عمل منقح النصوص من TEXT EDITOR ويحفظ البرنامج المكتوب بنفس الوقت، ولكن عندما يريد المبرمج حفظ برنامجه على وسط التخزين الثانوي دون أن ينهي العمل على البرنامج، فإنه يستخدم أمر SAVE لحفظ البرنامج مع الإبقاء على الشاشة كما هي عليه، أي يستطيع بعد تنفيذ هذا الأمر أن يستمر بالعمل على نفس البرنامج الشكل العام للأمر:

COMMAND INPUT > SAVE

و ينصح بعمل SAVE كلما انتهينا من كتابة صفحة أو كلما أدخلنا تعديلات كبيرة على البرنامج، لأنه في حالة وقوع خطأ في الحاسب وهو ما يسمى عادة CRASH فإنه من الممكن أن يخرب التعديلات الحديثة أو أن يلغي برنامجاً جديداً لم يعمل له حفظ SAVE قبل ذلك.

١٠ - أمر الإلغاء CANCEL

يمكن للمبرمج أن يصرف النظر عن التعديلات التي أدخلها على برنامجه لسبب أو لآخر، لذا فإن أمر CANCEL يلغي آخر تعديلات أدخلت على البرنامج قبل أن يعمل له SAVE أو قبل أن يخرج من البرنامج بـ PF15. والشكل العام للأمر:

COMMAND INPUT > CANCEL

ج - أمر البحث عن حرف أو سلسلة أحرف ، أمر التبديل :

FINDING AND CHANGING CHARACTER STRING

وهذان الأمران هما : أمر البحث FIND ويختصر بالحرف F
أمر التبديل CHANGE ويختصر بالحرف C

ويعتبر هاذان الأمران من الأوامر الأولية Primary Commands ، لذا يكتب في السطر الثاني من شاشة متفتح النصوص ، مقابل عبارة COMMAND INPUT

١ - أمر FIND

يستخدم هذا الأمر لاييجاد حرف أو سلسلة أحرف مكررة مرة أو عدة مرات في البرنامج - الملف -

والشكل العام للأمر هو:

COMMAND INPUT > FIND string - 1 [ALL]

العبارة ALL تفيد بالبحث عن السلسلة في كل البرنامج ابتداءً من السطر الأول حتى السطر الأخير. ويعطي عدد المرات المكررة في الزاوية اليمنى العليا من الشاشة. أما عندما تحذف عبارة ALL فانه يبدأ بالبحث عن أول تسلسل للأحرف المطلوبة مباشرة بعد الموقع الذي هو فيه، فإذا كان بالسطر ٥٠ يبدأ البحث من السطر ٥٠ إلى نهاية البرنامج - الملف - وكلما صادف هذه السلسلة المطلوبة أظهرها بلون غامق وقد وضعت الإشارة CURSOR تحتها وننتقل من سلسلة إلى سلسلة أخرى بضغط مفتاح PF17 لنرى نفس السلسلة في مكان آخر من البرنامج ، وعند انتهاء البرنامج - الملف - تظهر لدينا رسالة 'Bottom of Data Reached' في الزاوية اليمنى العليا من الشاشة. وعندما نضغط PF17 مرة ثانية تبدأ البحث من جديد على نفس السلسلة من أول البرنامج.

ويجب أن نراعي أنه إذا كانت السلسلة تحتوي على فراغاً SPACE أو فاصلة (،) COMMA ، أو كانت أحد عبارات الأوامر فيجب وضعها بين فاصلتين علويتين (') APOSTROPHE أو بين فاصلتين علويتين مكررتين (") QUOTES أمثلة :

COMMAND INPUT > F XYZ سلسلة بسيطة

COMMAND INPUT > F 'TO WORKI' فراغ سلسلة تحتوي على

COMMAND INPUT > F 'ALL' سلسلة تحتوي على عبارة
مستخدمة في الأوامر

في حال عدم وجود السلسلة String المطلوب البحث عنها في الملف كله تظهر رسالة
'Not Found' في الزاوية العليا اليمنى من الشاشة.

٢ - أمر CHANGE التبديل :

يستخدم هذا الأمر لإيجاد حرف أو سلسلة أحرف وتبديلها بسلسلة أحرف أخرى.
والشكل العام للأمر:

COMMAND INPUT > CHANGE String - 1 String - 2 [ALL]

أي بدل السلسلة ١ String - 1 بالسلسلة ٢ String - 2

ويمكن اختصار الأمر CHANGE إلى الحرف C*

إن عبارة All تفيد بأن يغير السلسلة ١ بالسلسلة ٢ في كل البرنامج أينما وجدت
هذه السلسلة.

مثال :

COMMAND INPUT > C WRK1 WRK2 ALL

أي بدل السلسلة من الأحرف 1 WRK بالسلسلة 2 WRK في كل البرنامج.

يمكن حذف عبارة ALL لأن استخدامها دقيق فهي قد تبديل سلسلة غير مطلوب
تبديلها بخطاءاً من المبرمج.

ويفضل استخدام الأمر CHANGE مع الأمر FIND بحيث يضغط PF17 من أجل
إيجاد السلسلة المطلوب تبديلها وعند التأكد من ذلك يضغط مفتاح PF18 من أجل تنفيذ
هذا التبديل، وهكذا...

(*) يجب الانتباه إلى أن الحرف C هو اختصار أمر CHANGE وهو أمر أولي، أما الأمر C هو طباعة سطر أمر يكتب
على السطر Command Line

أما السلسلة String 1 أو String 2 فتوضع بين فاصلتين علويتين إذا احتويتا أي فراغ أو فاصلة يرجع لأمر F — .

عند تبديل أي سلسلة بسلسلة أخرى تظهر أقصى يسار السطر الذي حصل فيه التبديل عبارة < CHG == أي تم التبديل.

ونذكر أنه يمكن تبديل سلسلة بسلسلة أحرف أطول منها أو أقصر. وفي حالة الأطول نجد أن الأمر يزيح الأحرف ليدخل الأحرف الزائدة، وفي حال عدم وجود فراغ مناسب يستوعب الأحرف الزائدة تظهر عبارة < ERR == في أقصى يسار السطر الجاري العمل عليه.

يمكن إزالة عبارة < CHG == أو < ERR == بتنفيذ أمر RESET كما مر معنا سابقاً.

كما ويمكن أن يأخذ الأوامر FIND و CHANGE صيغا معقدة أخرى، لكن يكفي بالشكل الحالي الذي يفى بالغرض.

انظر الشكل رقم (١١-٢٦) من أجل ملاحظة عمل أمري F و C. وكذلك الشكل رقم (١١-٢٧).

د — أوامر متقدمة :

ومنستعرض منها :

— أوامر سطرية — منها :

أمر الازاحة لليساار COLUMNS LEFT (')

أمر الازاحة لليمين COLUMNS RIGHT (')

— أوامر أساسية :

| | |
|--------|---------|
| COPY | PROFILE |
| MOVE | HEX |
| CREATE | |

١ - أمر الازاحة :

يستخدم هذا الأمر اعتباراً من السطر المطلوب بهدف إزاحة الأعمدة لليمين أو اليسار بقدر موضع أو أكثر وتستخدم لذلك الرموز التالية :

SHIFT RIGHT) الازاحة لليمين
SHIFT LEFT (الازاحة لليساار

يدخل هذا الأمر - الرمز - في المكان المخصص لأوامر الأسطر ويسمى Line Command Area وهو أقصى يسار السطر، ثم يضغط مفتاح ENTER فتتم الازاحة لليمين أو لليساار على حسب الرمز - الأمر - ، أما إذا أردنا الازاحة أكثر من موقع واحد لليساار نكتب n بحيث ترمز n إلى عدد الأعمدة المطلوب إزاحتها والصيغة العامة

(■

) n

كما ويمكن إزاحة مجموعة أسطر متتالية BLOCK OF LINES — أو برنامج كامل — لليمين أو لليساار باستخدام نفس الرموز وبالشكل التالي :

لوفرضنا أننا نريد إزاحة ٢٠ سطر لليساار بمقدار ٨ مواقع فإننا نكتب في السطر الأول من المجموعة الرمز 8)) وفي السطر الأخير من المجموعة ((— بدون رقم — ونضغط مفتاح ENTER لتتم الازاحة لليساار بمعدل ٨ مواقع 8 POSITIONS .

وتستخدم نفس الخطوات من أجل الازاحة لليمين لكن باستخدام الرمز *) .

انظر الشكل رقم ١١-٢٨ ، ١١-٢٩ ، ١١-٣٠ ، ١١-٣١ .

EDIT _ STD4253. TEST. COBOL (PROGA) _ 01.20

COLUMNS 001 072

COMMAND INPUT ==>C CARD-F CARDF

SCROLL ==> HALF

***** TOP OF DATA *****

000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. PROGA.
000300 * THIS PROGRAM PRINTS IPA _
000400 * NAME AND ADDRESS
000500 *
000600 ENVIRONMENT DIVISION.
000700 CONFIGURATION SECTION.
000800 SOURCE-COMPUTER. IBM-370.
000900 OBJECT-COMPUTER. IBM-370.
001000 *
001100 INPUT-OUTPUT SECTION.
001200 FILE-CONTROL.
001300 SELECT CARD-F ASSIGN TO UT-S-CARD.
001400 SELECT PRINT-F ASSIGN TO UT-PRFL.
001500 *
001600 DATA DIVISION.
001700 FILE SECTION.
001800 FD CARD-F RECORD CONTAINS 80 CHARACTERS
001900 LABEL RECORD IS OMITTED.
002000 01 CARD-REC PIC X (80).
002100 *

شکل رقم (۱۱-۲۶)

-۴۱۱-

EDIT _ STD4253. TEST. COBOL (PROGA) _ 01.20 _ CHARS 'CARD-F' CHANGED

COMMAND INPUT ==>

SCROLL ==> HALF

***** TOP OF DATA *****

000100 IDENTIFICATION DIVISION.

000200 PROGRAM-ID. PROGA.

000300 * THIS PROGRAM PRINTS IPA _

000400 * NAME AND ADDRESS

000500 *

000600 ENVIRONMENT DIVISION.

000700 CONFIGURATION SECTION.

000800 SOURCE-COMPUTER. IBM-370.

000900 OBJECT-COMPUTER. IBM-370.

001000 *

001100 INPUT-OUTPUT SECTION.

001200 FILE-CONTROL.

001300 SELECT CARD F ASSIGN TO UT-S-CARD.

001400 SELECT PRINT-F ASSIGN TO UT-PRFL.

001500 *

001600 DATA DIVISION.

001700 FILE SECTION.

001800 FD CARD F RECORD CONTAINS 80 CHARACTERS

001900 LABEL RECORD IS OMITTED.

002000 01 CARD-REC PIC X (80).

002100 *
شكل رقم (١١ - ٢٧)
بعد تنفيذ أمر التغيير

٢- أمر MOVE و COPY

يستخدم هذان الأمران كأوامر أولية أساسية PRIMARY COMMANDS، لذا يكتب في السطر الثاني مقابل عبارة .

.... COMMAND INPUT

و يستخدم هذان الأمران لدمج برنامج — ملف — جزئي مع برنامج — ملف — أساسي MERGE DATA ، ونعيد إلى الأذهان الفرق بين عبارتي COPY و MOVE إذ أن الأولى تنسخ نسخة لمكان آخر ويظل الأصل كما هو، بين الثانية — MOVE — يتم نقل المحتوى كله من مكان لآخر دون الاحتفاظ بأي أصل .

EDIT _ STD4253. TEST. COBOL (PROGA) _ 01.20 _ SCREEN IMAGE IS PRINTED

COMMAND INPUT ==>

SCROLL ==> HALF

***** ***** TOP OF DATA *****

000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. PROGA.
000300 * THIS PROGRAM PRINTS IPA_
000400 * NAME AND ADDRESS
000500 #
((200 ENVIRONMENT DIVISION.
000700 CONFIGURATION SECTION.
000800 SOURCE-COMPUTER. IBM-370.
((0900 OBJECT-COMPUTER. IBM-370.
001000 #
001100 INPUT-OUTPUT SECTION.
001200 FILE-CONTROL.
001300
001400 SELECT PRINT-F ASSIGN TO UT-PRFL.
001500 *
001600 DATA DIVISION.
001700 FILE SECTION.
001800 FD CARD-F RECORD CONTAINS 80 CHARACTERS
001900 LABEL RECORD IS OMITTED.
002000 01 CARD-REC PIC X (80).
002100 *

شکل رقم (۱۱-۲۸)

EDIT _ STD4253. TEST. COBOL (PROGA) _ 01.20 _ COLUMNS 001 072

COMMAND INPUT ==> SCROLL ==> HALF

***** TOP OF DATA *****

000100 IDENTIFICATION DIVISION
000200 PROGRAM-ID. PROGA.
000300 * THIS PROGRAM PRINTS IPA_
000400 * NAME AND ADDRESS
000500 *
000600 ENVIRONMENT DIVISION.
000700 CONFIGURATION SECTION.
000800 SOURCE-COMPUTER. IBM-370.
000900 OBJECT-COMPUTER. IBM-370.
001000 *
001100 INPUT-OUTPUT SECTION.
001200 FILE-CONTROL.
001300 SELECT CARD-F ASSIGN TO UT-S-CARD.
001400 SELECT PRINT-F ASSIGN TO UT-PRFL.
001500 *
001600 DATA DIVISION.
001700 FILE SECTION.
001800 FD CARD-F RECORD CONTAINS 80 CHARACTERS
001900 LABEL RECORD IS OMITTED.
002000 01 CARD-REC PIC X (80).
002100 *

شکل رقم (۱۱-۲۹)

EDIT _ STD4253, TEST: COBOL (PROGA) _ 01.20 _ SCREEN IMAGE IS PRINTED

COMMAND INPUT ==>

SCROLL ==> HALF

***** ***** TOP OF DATA *****

000100 IDENTIFICATION DIVISION.

000200 PROGRAM-ID. PROGA.

000300 * THIS PROGRAM PRINTS IPA _

000400 * NAME AND ADDRESS

000500 *

)0600 ENVIRONMENT DIVISION.

000700 CONFIGURATION SECTION.

000800 SOURCE-COMPUTER. IBM-370.

)0900 OBJECT-COMPUTER. IBM-370.

001000 *

001100 INPUT-OUTPUT SECTION.

001200 FILE-CONTROL.

001300 SELECT CARD-F ASSIGN TO UT-S-CARD.

001400 SELECT PRINT-F ASSIGN TO UT-PRFL.

001500 *

001600 DATA DIVISION.

001700 FILE SECTION.

001800 FD CARD-F RECORD CONTAINS 80 CHARACTERS

001900 LABEL RECORD IS OMITTED.

002000 01 CARD-REC PIC X (80).

002100 *

شکل رقم (۱۱-۳۰)

EDIT _ STD4253. TEST. COBOL (PROGA) _ 01.20 _ COLUMNS 001 072

COMMAND INPUT ==> SCROLL ==> HALF

```
***** TOP OF DATA *****
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. PROGA.
000300 * THIS PROGRAM PRINTS IPA _
000400 * NAME AND ADDRESS
000500 *
000600 ENVIRONMENT DIVISION.
000700 CONFIGURATION SECTION.
000800 SOURCE-COMPUTER. IBM-370.
000900 OBJECT-COMPUTER. IBM-370.
001000 *
001100 INPUT-OUTPUT SECTION.
001200 FILE-CONTROL.
001300 SELECT CARD-F ASSIGN TO UT-S-CARD.
001400 SELECT PRINT-F ASSIGN TO UT-PRFL.
001500 *
001600 DATA DIVISION.
001700 FILE SECTION.
001800 FD CARD-F RECORD CONTAINS 80 CHARACTERS
001900 LABEL RECORD IS OMITTED.
002000 01 CARD-REC PIC X (80).
002100 *
```

شکل رقم (۱۱-۳۱)

وتتم عملية الدمج حسب الخطوات التالية :

١ - يطلب البرنامج الأصلي المراد إضافة برنامج جزئي - ملف جزئي - إليه
بالخيار 2 EDIT

٢ - يحدد الموقع ضمن البرنامج الأصلي الذي سيدخل الجزء الإضافي إليه وذلك
بوضع الأمر A (After) أو B (Before) في السطر المناسب وبالمكان المناسب

Line Commands Area

٣ - يكتب في السطر الثاني مكان الأوامر الأساسية الأمر COPY أو MOVE
حسب احتياجات العمل ، وأمامه اسم الملف الذي سيدخل على البرنامج الأساسي .
وكما هو مبين :

COMMAND > INPUT COPY Filename



يكتب هنا اسم الملف
الجزئي المطلوب ادخاله

٤ - اضغط مفتاح ENTER لتنفيذ الأمر.

٣ - أمر REPLACE, CREATE

وهما من الأوامر الأساسية Primary Commands والشكل العام لهما :

COMMAND INPUT > $\left\{ \begin{array}{l} \text{CREATE} \\ \text{REPLACE} \end{array} \right\}$ Filename

ويستخدم هذان الأمران عندما نريد فصل أو تجزئة برنامج - ملف - إلى ملف
فرعي آخر.

مثال : لو أردنا فصل جزء من برنامج إلى برنامج فرعي نتبع ما يلي :

١ - نطلب البرنامج الأسامي - الأصلي - بالخيار 2 EDIT

٢ - نحدد الأسطر التي نريد فصلها نهائياً بأمر MM أو إذا أردنا نسخ صورة منها نستخدم الأمر CC وهما أمران على مستوى السطر Line Command.

٣ - نكتب في المكان المخصص لكتابة الأوامر الأساسية الأمر CREATE كما هو واضح.

COMMAND INPUT > CREATE Filename

↓
اسم الملف الجديد

الذي سيحتوي على الأسطر المنقولة

٤ - نضغط مفتاح ENTER من أجل تنفيذ هذا الأمر.

بعد تنفيذ هذا الأمر يتشكل لدينا ملف جديد يحتوي على الأسطر المنقولة أو المنسوخة من الملف الأساسي.

File name وهو اسم الملف فيجب ان يكون جديداً غير مستخدم مسبقاً
NOT EXIST عند استخدام أمر CREATE ولا يشترط ذلك عند استخدام أمر
REPLACE لأنه ولو كان هناك ملف – برنامج – بنفس الاسم فان أمر REPLACE
سوف يلغي القديم ويحل محله الجديد.

٥ - استخدام أمر PROFILE إطار الملف .

يمكن التعرف على حالة بعض الأوامر فيما إذا كانت في حالة عمل ON أو عدم العمل OFF وذلك بكتابة أمر PROFILE كتعليق أساسية فتظهر هذه المعلومات على الأسطر الثلاثة الأولى من الشاشة الشكل العام لها:

> PROFILE

انظر الشكل رقم (١١ - ٣٢)

```

EDIT _ STD4253. TEST. COBOL (PROGA) _ 01.20 _ COLUMNS 001 072
COMMAND INPUT ==> PROFILE SCROLL ==> HALF
***** TOP OF DATA *****
= PROF > ....COBOL (FIXED - 80) .... RECOVERY ON ... NUMBER ON STD .....
= PROF > .... CAPS ON .... HEX OFF .... NULLS OFF .... TABS OFF .....
= PROF > .... AUTONUM OFF .... PRINT OFF .... STATS ON .....

000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. PROGA.
000300 * THIS PROGRAM PRINTS IPA_
000400 * NAME AND ADDRESS
000500 *
000600 ENVIRONMENT DIVISION.
000700 CONFIGURATION SECTION.
000800 SOURCE-COMPUTER. IBM-370.
000900 OBJECT-COMPUTER. IBM-370.
001000 *
001100 INPUT-OUTPUT SECTION.
001200 FILE-CONTROL.
001300 SELECT CARD-F ASSIGN TO UT-S-CARD.
001400 SELECT PRINT-F ASSIGN TO UT-PRFL.
001500 *
001600 DATA DIVISION.
001700 FILE SECTION.
001800 FD CARD-F RECORD CONTAINS 80 CHARACTERS

```

شکل رقم (۱۱ - ۳۲)

٦ - استخدام أمر HEX

يستخدم هذا الأمر لتحويل الأحرف المكتوبة من الصيغة العامة STANDARD FORM إلى الصيغة السداسية عشر HEXADECIMAL والشكل العام للأمر:

> HEX ON / OFF

وعند وضع هذا الأمر على ON فإن كل سطر من سطر البيانات سوف يترجم إلى سطرين بالصيغة السداسية عشر HEXADECIMAL

ويمكن تبديل أي حرف بالصيغة السداسية عشر إلى حرف آخر بنفس الصيغة .
ويمكن الغاء هذه الحالة بكتابة HEX OFF

فتعود الكتابة إلى الشكل العام العادي .
انظر الشكل رقم (١١-٣٣) والشكل رقم (١١-٣٤) .

***** TOP OF DATA *****

```
000400      *  NAME AND ADDRESS
          44444444444444444444DCDC4CDC4CCDC4CCDCDCEE 44444444444444444444
          000000C000000000000015401540154014949522 00000000000000000000000000000000
```

EDIT _ STD4253. TEST. COBOL (PROGA) _ 01.20 _ COLUMNS 001 072

COMMAND INPUT ==> HEX OFF SCROLL ==> HALF

***** TOP OF DATA *****

000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. PROGA.
000300 * THIS PROGRAM PRINTS IPA_
000400 * NAME AND ADDRESS
000500 *
000600 ENVIRONMENT DIVISION.
000700 CONFIGURATION SECTION.
000800 SOURCE-COMPUTER. IBM-370.
000900 OBJECT-COMPUTER. IBM-370.
001000 *
001100 INPUT-OUTPUT SECTION.
001200 FILE-CONTROL.
001300 SELECT CARD-F ASSIGN TO UT-S-CARD.
001400 SELECT PRINT-F ASSIGN TO UT-PRFL.
001500 *
001600 DATA DIVISION.
001700 FILE SECTION.
001800 FD CARD-F RECORD CONTAINS 80 CHARACTERS
001900 LABEL RECORD IS OMITTED.
002000 01 CARD-REC PIC X (80).
002100 *

شکل رقم (۱۱-۳۴)

١١ - ٤ استخدام نظام الحاسب TI990

١١ - ٤ - ١ استخدام TEXT EDITOR منقح النصوص في جهاز TEXAS INSTRUMENTS

يستخدم منقح النصوص TEXT EDITOR كوسيلة لادخال ملفات، أو لتعديل ملفات قديمة و يستخدم عبر الشاشات TERMINALS يمكن استدعاء منقح النصوص هذا بتعليمة XE (EXECUTE EDIT)، ويمكن بعدها تعيين اسم الملف إذا كان الملف قديماً، أو ادخال ملف جديد كما سيرد شرحه فيما بعد. ويفترض في هذه الحالة أن المبرمج قد أدخل USER ID والـ PASSCODE الخاصة به.

عندما نريد استخدام TEXT EDITOR من أجل تعديل ملف قديم، ندخل اسم الملف و يظهر الملف على الشاشة ٢٤ سطراً مرقماً من ١ إلى ٢٤ ويجري إضافة أحرف جديدة أو حذف أحرف وإضافة أسطر والغاء أسطر لكن كل ذلك لا يتناول النسخة الأصلية للملف إنما نسخة ثانية تسمى ملف التعديل.

ننتهي من عملية التنقيح أي نخرج من TEXT EDITOR بتعليمة

QE (QUIT EDIT)

ويمكن للمبرمج أن يلغي كل التعديلات التي أدخلها أو أن يثبت هذه التعديلات ويمكن له أن يكتب الملف المعدل على ملف آخر ويحتفظ بالملف الأساسي بدون تعديل.

١١ - ٤ - برنامج جديد :

كيفية استخدام الشاشات في TEXT EDITOR ؟

يفترض أننا فتحنا الشاشة وذلك بإدخال الرمز التعريفي - وكلمة السر بعدها

نكتب عبارة XE فتظهر لدينا الشاشة التالية بعد ضغط مفتاح RETURN

INITIATE TEXT EDITOR

FILE ACCESS NAME : اسم البرنامج

نضغط RETURN دلالة على اننا نريد إنشاء ملف (برنامج) جديد فتظهر امامنا شاشة نظيفة عليها السطر التالي :

• EOF

وهذا معناه أن الملف فارغ ويحتوي فقط علامة نهاية الملف END OF FILE نضغط مفتاح التحكم F7 وكذلك المفتاح الرمادي غير المعنن وعندها ستتقل علامة EOF • للسطر الثاني وتبدأ بادخال برنامج سطرأ سطرأ ، وكلما انتهينا من كتابة سطر — سجل RECORD — أضغط ENTER ليأخذ السطر التالي آلياً .

ويمكننا خلال وجودنا في TEXT EDITOR استخدام أي أمر من أوامر التنقيح والتي سيرد ذكرها فيما بعد .

بعد الانتهاء من ادخال البرنامج ، نضغط مفتاح CMD ونكتب QE فتظهر لدينا الرسالة أو الشاشة التالية :

QUIT EDIT

NO : ? ABORT هل يخرب العمل ؟

هذا معناه إننا إذا أردنا الغاء ما كتبناه نكتب YES وبالتالي سيلغي الحاسب كل شيء كتبناه ، اما إذا أجبنا بـ NO فهذا معناه أنه سيحتفظ بما كتبناه سابقاً ، بعدها نضغط RETURN لتظهر أمامنا الشاشة التالية :

QUIT EDIT

OUT PUT FILE ACCESS NAME : اسم البرنامج

NO : ? REPLACE

MOD LIST ACCESS NAME :

OUT PUT ACCESS NAME يعني اسم البرنامج المراد تخزينه أي نكتب اسم البرنامج

والسطر الثاني REPLACE ، أي هل يبدل هذه النسخة بالنسخة السابقة ؟
في حالتنا هذه لا يوجد نسخة سابقة ، فلو ادخلنا الجواب YES أو NO (نعم أو لا)
لا فرق في ذلك .

MOD LIST ACCESS NAME

و يذكر هنا اسم الملف الذي يحتوي على التعديلات ، وكذلك في حالتنا هذه
لا يوجد تعديلات بل أن الملف جديد لذلك نضغط RETURN
بهذا يكون العمل على منقح النصوص انتهى وتم حفظ البرنامج على وسط التخزين
الثانوي .

١١ - ٤ - ٢ - تنقيح ملف موجود سابقاً في جهاز TI

إن أي برنامج يدخل للحاسب لابد وأنه يحتاج إلى تنقيح ، أما من أخطاء قواعدية
أو من أخطاء منطقية ، وأحياناً يتطلب التعديل ، تغيير في المعطيات الأساسية للبرنامج .

فتتسلسل عملية طلب الملف — البرنامج — للتعديل تتم وفقاً للخطوات التالية :

١ — نطلب TEXT EDITOR بالأمر XE

٢ — تظهر لدينا الشاشة التالية :

INITIATE TEXT EDITOR

FILE ACCESS NAME : اسم البرنامج

فتدخل اسم البرنامج الموجود أصلاً في الحاسب ، فيظهر أمامنا البرنامج على الشاشة
اعتباراً من السطر الأول ولغاية السطر ٢٤ وأمام كل سطر يوجد رقم السطر المتسلسل .

يمكننا طلب الصفحات التالية للبرنامج بضغط مفتاح F1 أو يمكننا الرجوع
صفحات للوراء بضغط مفتاح F2 كما يمكننا استخدام الاسهم ← → ↑ ↓ لتحريك
الاشارة المضيئة CURSOR ووضعها على الأحرف المطلوبة وتعديل هذه الأحرف .
كما يمكننا استخدام مفاتيح DEL CHAR لحذف أحرف معينة أو مفتاح
INS CHAR لاضافة أحرف معينة بين سلسلة من الأحرف ضمن السطر الواحد . كما
يمكننا حذف السطر وإضافة السطر جديدة للملف — البرنامج —
عند الانتهاء من التعديل اضغط مفتاح CMD البرقالي ونكتب QE فنظهر أمامنا
الشاشة التالية :

QUIT EDIT

ABORT ? NO

إذا كنا لا نرغب في التعديلات التي عملنا عليها فنكتب YES وبالتالي يرجع
الملف — البرنامج — كما كان أصلاً دون أي تعديل وكأن شيء لم يكن . أما إذا كنا
جادين في التعديل فنكتب NO وبالتالي ستظهر لدينا الشاشة التالية :

QUIT EDIT

OUTPUT FILE ACCESS NAME :

REPLACE ? : YES

MOD LIST ACCESS NAME :

عند عبارة OUTPUT FILE ACCESS NAME يجب أن نضع اسم البرنامج الذي
كنا نعمل عليه ، أو اسم البرنامج — ملف — جديد نرغب في إنشائه وبالتالي يكون
عندنا برنامجين الأصل والنسخة المعدلة ، لكن في الأحوال العادية نستخدم نفس اسم
البرنامج في الأول وفي النهاية

FILE ACCESS NAME : TEST أي عند البداية

OUTPUT FILE ACCESS NAME : TEST وعند النهاية

أما عبارة REPLACE فهي تعني أن النسخة الجديدة المنقحة ستحل محل النسخة الأصلية إذا كان الجواب YES ، أما إذا كان الجواب NO فتعود للبرنامج وتكتب QE من جديد وتعين اسم ملف الاخراج جديد :

OUTPUT FILE ACCESS NAME

أما عبارة MODE LIST ACCESS NAME فهو اسم ملف يحوي على الأسطر المعدلة وأصل هذه الأسطر، ويمكن إهمالها هنا واضغط فقط RETURN لتعود للشاشة الأصلية للحاسب .

استخدام بعض أوامر TEXT EDITOR

يمكن تلخيص بعض أوامر منقح النصوص كما يلي :

الأمر

| | |
|----|---------------------------------|
| CL | طباعة أسطر COPY LINES |
| DL | حذف أسطر DELETE LINES |
| ML | تحريك أسطر MOVE LINES |
| DS | حذف سلسلة أحرف DELETE STRING |
| RS | تغيير سلسلة أحرف REPLACE STRING |
| FS | إيجاد سلسلة أحرف FIND STRING |
| MT | تغيير مكان الأعمدة MODIFY TAB |
| SL | البحث عن أسطر SHOW LINES |

• كل هذه الأوامر تنفذ بعد ضغط مفتاح CMD

طباعة أسطر CL : COPY LINES

يستخدم هذا الأمر من أجل طباعة — نسخ — سطر أو أكثر من مكان معين إلى مكان آخر وعند كتابة هذا الأمر تظهر الشاشة التالية :

COPY LINES

رقم سطر البداية START LINE :

رقم سطر النهاية END LINE :

INSERT AFTER LINE : رقم السطر الذي بعده

سيتم ادخال الأسطر الجديدة

حذف أسطر DL DELETE LINES

و يستخدم هذا الأمر لحذف سطر أو مجموعة أسطر، وعند كتابة هذا الأمر تظهر لدينا الشاشة التالية :

DELETE LINES

رقم سطر البداية المراد حذفه START LINE :

رقم سطر النهاية المراد حذفه END LINE :

يتم حذف الأسطر من رقم البداية إلى رقم النهاية وما بينهما .

تحريك أسطر ML MOVE LINE

يستخدم هذا الأمر لتحريك سطر أو عدة أسطر من مكان في الملف لكان آخر في الملف — البرنامج — وعند كتابة هذا الأمر تظهر لدينا الشاشة التالية :

MOVE LINES

رقم بداية الأسطر المراد تحريكها START LINE :

رقم نهاية الأسطر المراد تحريكها END LINE :

INSERT AFTER LINE : رقم السطر المراد ادخال الأسطر بعده

حذف سلسلة أحرف **DS DELETE STRING**

هذا الأمر يحذف سلسلة من الأحرف يعينها المبرمج وعند كتابة هذا الأمر تظهر لديك الشاشة التالية :

DELETE STRING

عدد مرّات تكرار **NUMBER OF OCCURENCES :**
السلسلة المراد حذفها
عمود البداية **END COLUMN :**
عمود النهاية **START COLUMN :**
السلسلة **STRING :**

إيجاد سلسلة أحرف **FS FIND STRING**

يستخدم هذا الأمر من أجل البحث عن سلسلة أحرف يعينها المبرمج وعند كتابة هذا الأمر تظهر لك الشاشة التالية :

FIND STRING

رقم السلسلة المراد تعيينها **OCCURENCE NUMBER :**
رقم عمود البداية **START COLUMN :**
رقم عمود النهاية **END COLUMN :**
السلسلة **STRING :**

تغيير مجموعة أحرف بمجموعة ثانية **RS REPLACE STRING**

يستخدم هذا الأمر لإبدال سلسلة أحرف مسلسلة بأحرف ثانية يعينها المبرمج وعند كتابة هذا الأمر تظهر لديك الشاشة التالية :

REPLACE STRING :

NUMBER OF OCCURENCES : عدد المرات المراد

إبدال السلسلة بسلسلة الثانية

عمود البداية

START COLUMN :

عمود النهاية

END COLUMN :

السلسلة الأصلية

STRING :

السلسلة البديلة .

CHANGE :

تعديل أعمدة . MT MODIFY TAB

يستخدم هذا الأمر لتعديل الأعمدة التي ستقفز عنها الإشارة المضيئة CURSOR ،
وعند كتابة هذا الأمر تظهر لديك الشاشة التالية :

MODIFY TAB

TAB COLUMNS : يدخل رقم الأعمدة التي يريد للاشارة

المضيئة المروور عليها و بينها فاصلة (,)

أظهر سطر رقم SL SHOW LINE

يستخدم هذا الأمر من أجل وضع أو أظهار رقم السطر الموجود في هذا الأمر على
الشاشة ، و يمكن وضع الحرف B لظهار أسطر البداية ، أو الحرف E لظهار أحرف
النهاية أو أي رقم سطر.

وعند كتابة هذا الأمر يظهر لديك الشاشة التالية :

SHOW LINE

LINE : رقم السطر

الفصل الثاني عشر

ترجمة البرنامج وتنفيذه

COMPILE AND EXECUTE COBOL PROGRAM

١٢ - ١ - لمحة عامة :

ستتحدث في هذا الفصل : أهمية المترجم واستخدامه في نوعين من الأجهزة المتوفرة في المعهد : الأول الحاسب I.B.M.3033 والثاني الحاسب I.B.M.990 .
ولابد من التذكير بأن المترجم هو من البرامج الأساسية SOFTWARE الموجودة في كل حاسب يستخدم لغة كوبول .
ويعتلف المترجم من حاسب لآخر طبقاً لنظام التشغيل المستخدم ومساحة الذاكرة الرئيسية المتوفرة ولكن تبقى الاختلافات ليست جوهرية بالنسبة للغة ككل ، ونؤكد بنصح المبرمجين بالرجوع إلى مراجع الشركة التي تشرح وتبين إمكانيات المترجم المتوفر مع الحاسب .

١٢ - ٢ - ترجمة وتنفيذ البرنامج باستخدام

الحاسب IBM3033 / نظام التشغيل MVS

- إن أهمية المترجم ترجع إلى أنه يقوم بترجمة اللغة العليا — في حالتنا هذه الكوبول — إلى لغة الآلة و يصيغها بصيغة التنفيذ OBJECT
ويمكن عمل ترجمة للبرنامج بعدة طرق منها :
١ — استخدام JCL خاص وتنفيذه بأمر SUBMIT
٢ — استخدام FOREGROUND أي الخيار 4.2 والكشف عن النتائج

بـ USER-ID. LIST

٣ - استخدام BACKGROUND أي الخيار 5.2 والكشف عن الأخطاء
بالخيار 3.8

وسنستعرض هنا الطريقة رقم ٣ لبساطتها وسهولة التعرف على الأخطاء .

استخدام BACKGROUND الخيار 5.2 لعمل ترجمة COMPILE للبرنامج :
اعتباراً من الشاشة الأساسية رقم ١١-١ نطلب الخيار 5 فتأتي الشاشة رقم
١٢-١ نعدل بطاقات العمل JCL كما هو وارد في الشكل .

بعدها نكتب الخيار 2 من هذه الشاشة مقابل عبارة SELECT OPTION انظر
الشاشة رقم ١٢-١ .

بعد ان نضغط مفتاح ENTER تأتي الشاشة رقم ١٢-٢ نكتب اسم البرنامج
المطلوب عمل ترجمة له ونضغط مفتاح ENTER عندها تعود الشاشة رقم ١٢-٣ نضغط
PF15 أي ENDKEY حتى ينفذ العمل وإذا أردنا الغاء كتبنا عبارة CANCEL

نقرأ اسم العمل الذي تم تنفيذه وهو عبارة عن الرقم التعريفي USER-ID الخاص
مضافاً إليه حرف (من A إلى Z) أو رقم ، وهذا الاسم هو الذي ستطلبه عند التعرف
على الأخطاء في الخيار 3.8.

BACKGROUND SELECTION MENU

SELECTION OPTION ==>

- | | |
|------------------------------|--------------------------------|
| 1 _ SYSTEM ASSEMBLER | 4 _ PL / I CHECKOUT COMPILER |
| 2 _ OS / VS COBOL COMPILER | 5 _ PL / I OPTIMIZING COMPILER |
| 3 _ FORTRAN IV (G1) COMPILER | 6 _ LINKAGE EDITOR |
| | 9 _ PASCAL / VS COMPILER |

JOB STATEMENT INFORMATION : (VERIFY BEFORE PROCEEDING)

==> // STD,XXX^oY JOB (STD,XXX), 'NNNN',

==> // CLASS = H, MSGCLASS = Y,

==> // MSGLEVEL = (1,1), NOTIFY = STDXXXX

==> // •

==> // •

الشاشة رقم (١٢-١)

• الحرف Y : يقصد به أي حرف من A إلى Z

BACKGROUND COBOL COMPILE

ENTER / VERIFY PARAMETERS BELOW :

PROJECT ==> STDXXXX

LIBRARY ==> TEST ==> ==> ==>

TYPE ==> COBOL

MEMBER ==>

LIST ID ==> (BLANK FOR HARDCOPY LISTING)

SYSOUT CLASS ==> Y (IF HARDCOPY REQUESTED)

COMPILER OPTIONS :

TEST ==> NOTEST (TEST OR NOTEST)

OTHER ==> SOURCE

شاشة رقم (١٢-٧)

-٤٣٦-

_____ BACKGROUND SELECTION MENU _____ JCL GENERATED

SELECTION OPTION ==>

- | | |
|------------------------------|--------------------------------|
| 1 _ SYSTEM ASSEMBLER | 4 _ PL / I CHECKOUT COMPILER |
| 2 _ OS / VS COBOL COMPILER | 5 _ PL / I OPTIMIZING COMPILER |
| 3 _ FORTRAN IV (G1) COMPILER | 6 _ LINKAGE EDITOR |
| | 9 _ PASCAL / VS COMPILER |

ENTER OPTION TO CONTINUE GENERATING JCL

ENTER CANCEL ON OPTION LINE TO EXIT WITHOUT SUBMITTING JOB

PRESS END KEY TO SUBMIT JOB

JOB STATEMENT INFORMATION : (VERIFY BEFORE PROCEEDING)

==> // STD,XXXXY JOB (STD,XXXX),'NNNN',

==> // CLASS = H,MSGCLASS = Y,

==> // MSGLEVEL = (1,1), NOTIFY = STD XXXX

==> // *

==> // *

الفاشة رقم (١٢ - ٣)

الحرف ٢ : أي حرف من A إلى Z

الكشف عن الأخطاء الناتجة من عمل COMPILE في الخيار 5.2 :

بعد الانتهاء من مرحلة الترجمة لابد وان نجد طريقة لاستعراض ناتج الترجمة، إذ من النادر جداً كتابة برنامج دون أخطاء قواعدية أو منطقية .

و يتم الكشف عن ناتج عملية الترجمة COMPILE والذي استخدمنا فيها الخيار 5.2 بكتابة الخيار 3.8 اعتباراً من الشاشة الأساسية فتظهر الشاشة رقم ١٢-٦ نضع الاسم التعريفي أي اسم العمل JOB-ID والذي أخذناه مسبقاً من الخيار 5.2 مقابل العبارة
JOB NAME

ننتظر قليلاً ونضغط مفتاح ENTER فتظهر لنا عبارات JCL بعدها نضغط PF20 لنستعرض البرنامج وفي نهاية البرنامج نجد الأخطاء ورقم الأسطر الموجودة فيها .

لا يمكن تصحيح هذه الأخطاء هنا في الخيار 3.8 بل يفضل طباعة هذا العمل على الطباعة إذا كانت الأخطاء كثيرة أو عمل SPLIT للشاشة أي فصل الشاشة باستخدام PF14 وطلب نفس البرنامج بالخيار 2 EDIT وتصحيح هذه الأخطاء و يستمان بالمفتاح PF21 للانتقال من شاشة إلى أخرى .

من الممكن طبعاً وهو الأسهل الخروج من الخيار 3.8 بعد طباعته، أو بعد حفظ الأخطاء أو تدوينها على الورق، ثم الدخول إلى برنامج EDIT وطلب البرنامج المراد العمل عليه .

يفضل عند الانتهاء من استعراض أي عمل حذف ذلك العمل بوضع الحرف D في SELECT OPTION وذلك لعدم تراكم الأعمال مما يشغل حيز كبير على القرص وكذلك تتكرر أسماء الأعمال مما يجعل استعراضها صعباً .

UTILITY SELECTION MENU

SELECT OPTION ==>

- 1 LIBRARY -- LIBRARY UTILITY :
PRINT INDEX LISTING OR ENTIRE DATASET
PRINT, RENAME, DELETE, OR BROWSE MEMBERS
COMPRESS DATASET
- 2 DATASET -- DATASET UTILITY :
DISPLAY DATASET INFORMATION
ALLOCATE, RENAME, OR DELETE ENTIRE DATASET
CATALOG OR UNCATALOG DATASET
- 3 MOVE COPY -- MOVE OR COPY MEMBERS OR DATASETS
- 4 CATALOG -- CATALOG MANAGEMENT :
DISPLAY OR PRINT CATALOG ENTRIES
INITIALIZE OR DELETE USER CATALOG ALIAS
- 5 RESET -- RESET STATISTICS FOR MEMBERS OF SPF LIBRARY
- 6 HARDCOPY -- INITIATE HARDCOPY OUTPUT
- 7 VTOC -- DISPLAY OR PRINT VTOC ENTRIES FOR A DASD VOLUME
- 8 OUTLIST -- DISPLAY, DELETE, OR PRINT HELD JOB OUTPUT
- 9 SCRIPT/VS -- FORMAT, DISPLAY, AND OPTIONALLY PRINT SCRIPT TEXT

شکل رقم (۱۲-۱)

BACKGROUND LINKAGE EDIT

ENTER/VERIFY PARAMETERS BELOW :

PROJECT ==> STD XXXX

LIBRARY ==> TEST ==> ==> ==>

TYPE ==> OBJ

MEMBER ==>

اسم البرنامج

LIST ID ==> (BLANK FOR HARDCOPY LISTING)

SYSOUT CLASS ==> Y (IF HARDCOPY REQUESTED)

LINKAGE EDITOR OPTIONS :

TERM ==> (TERM OR BLANK)

OTHER ==> LET, LIST, MAP

COMPILER SUBROUTINE LIBRARIES : (IN APOSTROPHES)

==>

==>

شكل رقم (١٢-٥)

OUTLIST UTILITY

SELECT OPTION ==>

L — LIST JOB NAMES/ID'S VIA THE TSO STATUS COMMAND لاظهار الأعمال قيد التنفيذ

D — DELETE JOB OUTPUT FROM SYSOUT HOLD QUEUE حذف نتيجة عمل منفذ

P — PRINT JOB OUTPUT AND DELETE FROM SYSOUT HOLD QUEUE

طباعة نتيجة عمل على الطابعة المركزية وحذفه

R — REQUEUE JOB OUTPUT TO A NEW OUTPUT CLASS طباعة نتيجة عمل على الطابعة المحلية

BLANK — DISPLAY JOB OUTPUT لاظهار نتيجة عمل على الشاشة

FOR JOB TO BE SELECTED :

JOB NAME ==> STDXXXXY

CLASS ==>

JOBID ==>

FOR JOB TO BE REQUEUED :

NEW OUTPUT CLASS ==> M

FOR JOB TO BE PRINTED :

(A FOR ANSI)

PRINTER CARRIAGE CONTROL ==> A (M FOR MACHINE)

(BLANK FOR NONE)

شكل رقم (١٢-٦)

* ملاحظة :

- ١ - يمكن في المرات التالية ان نكتب مباشرة الخيار 5.2 فتظهر الشاشة ١٢-٢ ثم نكتب اسم البرنامج ونضغط ENTER وبالتالي نأخذ اسم العمل JOB-ID دون الحاجة لكتابة عبارات JCL مرة ثانية.
- ٢ - قد يحصل في المرات الأولى أن يظهر JCL ERROR أخطاء في لغة التحكم، فيرجع للخيار 5 لتصحيح العبارات.

ربط البرنامج LINK باستخدام الخيار 5.7 :

بعد الانتهاء من تنقيح الأخطاء القواعدية من البرنامج، يحتاج هذا البرنامج للتنفيذ ويتم ذلك بعمل LINK لمجموعة البرامج الجزئية ووضع الناتج بملف DATASET تدعى LOAD، ولكل برنامج له صيغة LOAD صيغة التحميل من أجل التنفيذ، ويتم ذلك بكتابة الخيار 5.7 فتظهر الشاشة رقم (١٢-٥) نكتب اسم البرنامج مقابل للرمز شكل (١٢-٥) MEMBER من أجل ربط هذا البرنامج وتهيئته للتنفيذ. وكما مر معنا سابقاً تعطي هذه الخطوة اسم JOB NAME، يرجع للخيار 3.8 من أجل استعراض ناتج عملية الربط، فلو كانت المحصلة أو رمز الشرط RETURN CODE هو الصفر - وهو مسجل في بداية العمل - فهذا يعني أنه تم ربط البرنامج بنجاح، وإلا نراجع JCL الخاص بالخيار رقم 5 أو نراجع البرنامج من أجل التأكد من خلوه من الأخطاء.

ويمكن الرجوع إلى المرجع MANUAL الخاص بالأخطاء لايجاد مصدر الخطأ.

تنفيذ البرنامج EXECUTE

بعد الانتهاء من ربط البرنامج بنجاح، يصبح البرنامج جاهز لعملية التنفيذ، وتتم عملية التنفيذ بكتابة عدة أوامر من لغة التحكم JCL كما هو واضح بالشكل التالي :


```
// STDXXXX JOB (STD, XXXX), 'NNNN', CLASS = H, MSGLEVEL = (1, 1),
//      MSGCLASS = Y, NOTIFY = STDXXXX
// STPI      EXEC PGM = program name
// STEPLIB   DD  DSN = dataset, LOAD, DISP = SHR
// Input file name DD  DSN = dataset name, DISP = SHR
// Out put file name DD  DSN = dataset name, DISP = SHR
//
```

تدخل هذه الأسطر في الخيار 2 تحت اسم معين و يفضل أن يحتوي على الأحرف JCL ورمز عن اسم البرنامج الذي سينفذ لهولة الرجوع إليه .

مع مراعاة ما يلي :

١ - الأسطر تبدأ بالموقع ١ ولا تتبع قواعد لغة الكوبول ، بل تتبع قواعد لغة التحكم JCL .

٢ - تبديل XXXX بالرقم التعريفي ، و NNNN باسم المبرمج .

٣ - إن عبارة Program name تدل على اسم البرنامج الذي سينفذ والذي تمت عملية ربطه LINKED في الخطوة السابقة في الخيار 5.7

٤ - يذكر في عنوان الفقرة STEPLIB ، اسم الملف الحاوي على الملف LOAD فغالباً ما يكون

DSNAME = USERID. TEST. LOAD

٥ - إن كلا من Input file name و Output file name هما أسماء الملفات المذكورين في عبارة ASSIGN من SELECT ، فلو كان عندي المثال التالي :

SELECT INPUT F ASSIGN TO JFS - CARDF

فان الاسم CARDF هو الذي يجب أن يذكر في JCL أي

// CARDF DD DSN = USERID - TEST. CARDF, DISP = SHR...

أو أي اسم يعطى من قبل المشرف

نفس الكلام ينطبق على OUTPUT - FILE

وتجدر الإشارة أنه من المحتمل أن يكون في البرنامج أكثر من ملف ادخال وأكثر من ملف اخراج. فيجب أن يكون له عبارة DD (DATA DEFINITION) خاصة.

كما ويمكن الآ يحتوي البرنامج على ملف ادخال، أو لا يحتوي على ملف اخراج، لكن المهم أنه لكل ملف يجب أن يكون له عبارة DD خاصة به. أما عن اسم الملف الموجود على الوسط الثانوي فيعطى من قبل المشرف أو المسؤول عن تنظيم الملفات.

٦ - في حال كون ملف الاخراج على الطابعة يكون شكل عبارة DD كالتالي :

// Out put file name DD SYSOUT = x

يوضع حرف يدل على نوع الطابعة المستخدمة و يرجع للمشرف

٧ - قد تدخل البيانات مباشرة كبطاقات بعد بطاقة تعريف الملف و يكون شكلها كالتالي :

| | |
|---------------|--------------------------------|
| // CARDF DD * | وتدل النجمة (*) على أن السجلات |
| بطاقات { | CARDF التابعة للملف |
| البيانات { | تلي هذا السطر مباشرة |
| DATA { | |

٨ - من أجل زيادة المعلومات عن لغة التحكم بالعمل JCL يرجع للمرجع

OS/VS2 MVS JCL رقم GC28/0692/4

٩ - بعد الانتهاء من وضع كل المعايير المطلوبة نكتب الأمر SUBMIT في السطر الثاني من الشاشة مقابل عبارة INPUT COMMAND نضغط ENTER فيرسل العمل للتنفيذ و يعطى اسم عمل JOBNAME معين وغالباً ما يكون الاسم والرقم التعريفي USERID مضافاً إليه الحرف S - يمكن ان يكون غير S - ، ونخرج من برنامج EDIT بواسطة مفتاح PF15 عدة مرات .

الكشف عن ناتج التنفيذ :

بعد الانتهاء من التنفيذ، نذهب للخيار 3.8 للكشف عن ناتج العمل نكتب اسم العمل الذي أخذناه من الخطوة السابقة، ونستعرضه فإذا لم يحتوي على أخطاء قواعدية في JCL فهذا يعني أن البرنامج قد نفذ بنجاح، فإذا كان الناتج على الطابعة مثلاً نكتب الخيار P أو L من أجل تنفيذ الطابعة، أما إذا كان الناتج على قرص ممغنت، فيمكن استعراض الناتج في خيار آخر هو الخيار (1) BROWSE - اعتباراً من الشاشة الأساسية رقم ١١ - ١ .

إذا كان ناتج البرنامج OUTPUT يحوي على أخطاء منطقية أي ليست النتيجة المرجوة فهذا يعني خطأ في البرنامج، فيرجع للرسم الانسيابي FLOWCHART ويدرس و يعدل إذا لزم الأمر، ثم إلى صفحات الترميز CODING فتعدل، ثم يعدل البرنامج الموجود في الحاسب باستخدام منقح النصوص TEXT EDITOR، ويعمل له ترجمة ثم ربط ثم تنفيذ و يدرس الناتج من جديد. فإذا كان خالياً من الأخطاء المنطقية فهذا يعني أن البرنامج قد حقق المطلوب منه وإلاّ تباد الخطوات السابقة من جديد.

* توثيق البرنامج :

بعد الانتهاء من تنفيذ البرنامج والتأكد من النتائج ومناقشتها مع محلل النظم أو المشرف وإقرارها، يطبع نسخة من البرنامج — أو عدة نسخ — وذلك بالخيار 3.6 أو 3.8، وتحفظ نسخة من البرنامج مع المخطط الانسيابي له FLOWCHART مع أوراق الترميز CODING SHEET بالإضافة لنماذج ادخال أو نماذج اخراج أو نتائج في ملف بلاستيكي و يعطى له عنوان دال على محتواه.

و يفضل أن يحتوي البرنامج على ملاحظات COMMENTS تشير إلى خطوات العمل كما و يفضل كتابة الخطوات التي اتبعت لحل هذا البرنامج وتحفظ في ملف البرنامج و يُسلم إلى محلل النظم المسؤول أو مدير المشروع أو المشرف عند الطلب.

١٢ - ٣. ترجمة وتنفيذ البرنامج باستخدام الحاسب TI990

نظام التشغيل DX10

يستدعى المترجم COMPILER في الحاسب TI990 بكتابة الأمر التالي :

X CCF =

EXECUTE COBOL COMPILER FOREGROUND

تظهر لدينا الشاشة التالية :

SOURCE ACCESS NAME : اسم البرنامج الذي نرى

OBJECT ACCESS NAME : عمل له COMPILE

LIST ACCESS NAME :

OPTIONS : RETURN اضغط

PRINT WIDTH : 80 RETURN اضغط

• قد تحتوي بعض مراكز الكمبيوتر على نماذج توثيق خاصة ، فتستعمل عندئذ حسب توجيهات كل مركز.

اضغط RETURN حجم الصفحة 55 : PAGE SIZE

اضغط RETURN 1000 : PROGRAM SIZE (LINE)

يكتب في السطر الأول اسم البرنامج المراد : ترجمته وفي حالتنا :

• اسم البرنامج STDOOL, COBOL 3, SRC.

وفي السطر الثاني اسم البرنامج بشكل التنفيذ وفي حالتنا هنا :

اسم البرنامج STDOOL, COBOL 3, OBJ.

وفي السطر الثالث اسم الملف الذي سيوضح فيه قائمة بالبرنامج وبال عناوين المستخدمة وبالأخطاء إن وجدت والتي سيرجع إليها المبرمج لتحري الأخطاء وفي حالتنا هذه يكون :

اسم البرنامج STDOOL, COBOL 3, LST.

أما بقية الأسطر فهي عرض سطر الطباعة . وعدد الأسطر في الصفحة ، وحجم البرنامج بالأسطر فكلها جاهزة فضغط RETURN فقط .

ملاحظة هامة :

إن صيغة الأصل SOURCE هي التي تعدل فقط

أما صيغة التنفيذ OBJECT فليس للمبرمج عمل عليها .

أخيراً صيغة الـ LIST هي فقط لاستعراض الأخطاء ولا يعدل عليها .

تنفيذ البرنامج على الحاسب TI990

يتم تنفيذ البرنامج بعد عمل COMPILATION له وتشكيل OBJECT ، إن

صيغة التنفيذ OBJECT هي وحدها التي تنفذ فقط دون صيغة الأصل أو الـ LIST

- أسماء الملفات والوحدات يمكن أن تبدل من موقع لآخر ، فراجع للأستاذ المشرف من أجل أخذ الأسماء المطابقة .

فيعد التأكد من خلو البرنامج من الأخطاء القواعدية يشرع في تنفيذ البرنامج علينا تجهيز الملفات FILES ويتم بالشكل التالي :

وضمن برنامج الكوبول هناك لكل ملف في قسم بيئة البرنامج ENVIRONMENT DIVISION هناك تعليمة اختيار الملفات وهي :

SELECT file name ASSIGN TO

| |
|--------|
| INPUT |
| OUTPUT |
| RANDOM |
| PRINT |

 , "SYN"

إن ما بين الفاصلتين العلويتين هناك الأسم المختصر ويدعي SYNONAM هذا الأسم هو صلة الوصل بين البرنامج والوسط الموجود عليه الملف هل هو موجود على CARD READER أو TAPE أو PRINTER أو DISK

فقبل الشروع بتنفيذ البرنامج نعين الأسم المختصر بتعليمة

AS ASSIGN SYNONYM

نظهر لنا الشاشة التالية :

ASSIGN SYNONYM

SYNONYM الأسم المختصر الموجود في البرنامج ضمن الفاصلتين العلويتين VALUE : قيمة هذا الأسم أي على أي وسط موجود ؟

مثال : لو كان عندي ملف طباعة يكون توظيفه في ملف البرنامج كالتالي :

SELECT PRFL ASSING TO PRINT, "PRT". برنامج الكوبول

فالرمز PRT هو صلة الوصل بين البرنامج والطابعة فنكتب الاسم المختصر

ASSIGN SYNONYM

كالتالي :

SYNONYM : PRT الاسم الموجود بالبرنامج

VALUE : ST01 اسم الطابعة

إذا فالاسم PRT يعادل أو يدل على الطابعة التي اسمها ST01 مثلاً .
مثال : لو كان عندي ملف موجود على القرص الممغنط DISC أكتب :
SELECT DSCFL ASSIGN TO INPUT, "DFL" في برنامج الكوبول
ونشير إليه بالاسم المختصر كالتالي :

ASSIGN SYNONAM
SYNONAM : DFL الاسم المختصر الموجود في
البرنامج
VALUE : STD001. COBOL 3. FILES. TOFL الاسم الموجود على الديسك
بعد تعيين ال SYNONAM نكتب التعليمة التالية لتنفيذ البرنامج :
XCPF
EXECUTE COBOL PROGRAM FOREGROUND
فتظهر لنا الشاشة التالية :

OBJECT ACCESS NAME :
DEBUG MODE : NO RETURN اضغط
MESSAGE ACCESS NAME : RETURN اضغط
SWITCHS : 00000 RETURN اضغط
FUNCTION KEYS : NO RETURN اضغط
نكتب اسم البرنامج في السطر الأول ، و يشترط أن يكون اسم البرنامج بصيغة
التنفيذ أي في حالتنا هنا : اسم البرنامج STD001. COBOL 3. OBJ.
ونضغط RETURN لباقي الأسطر
ننتظر قليلاً لبدء تنفيذ البرنامج ، في حال ظهور رسائل تدل على الخطأ استدع
الأستاذ المشرف ، أما في حال حصول على نتائج خاطئة نرجع للبرنامج ونعدله بعد
دراسة الخطأ ونعاود ترجمته COMPILE ثم تنفيذه وهكذا

القسم الرابع

تطبيقات عملية

تطبيقات عملية تغطي العمليات الأساسية
في لغة البرمجة كوبول

الفصل الثالث عشر

المراجع

الفصل الرابع عشر

ملحقات

الفصل الخامس عشر

الفصل الثالث عشر

تطبيقات عملية للعمليات الأساسية في لغة البرمجة كوبول

١٣ - ١ ملحة عامة :

يحتوي هذا الفصل على التطبيقات الأساسية التي تفيد المبرمج أو الدارس في وضع وتصميم البرامج في المستقبل أو يمكن لهذه التطبيقات أن تعطي صورة أساسية لمختلف العمليات التي تنفذ على الملفات والبيانات باستخدام كلا النوعين من الحاسبات الصغيرة والكبيرة منها أي باستخدام الشاشات المتلفة والتعليمات المتطورة الخاصة بها وباتباع أسلوب التشغيل المعروفين: المباشر وغير المباشر حتى يتمكن المبرمج من الانطلاق في تطوير وتصميم برامج أكثر تعقيداً وأشمل معالجة، وليست هذه البرامج سوى نماذج عملية لتخير له الطريق للتقدم والتطور وفهم هذه اللغة واستخدامها من الناحية العملية بصورة أفضل.

١٣ - ٢ برنامج طباعة تقرير:

التمرين التالي وهو طباعة ملف متسلسل SEQUENTIAL مفروز على رقم المشترك D-No

إن عمليات الطباعة من العمليات السهلة نسبياً، والعنصر الأساسي فيها هو توصيف أسطر الطباعة وكما نعلم يشمل على ٣ أجزاء :

أ - العنوان PAGE HEADING

ب - جسم التقرير PAGE BODY

جـ - أسفل الصفحة PAGE FOOTING

أما بالنسبة للعناوين فتوصف في فصل التخزين الداخلي للبرنامج
WORKING STORAGE SECTION بمعدل سطر توصيف لكل سطر موجود على
ملف شكل الطباعة المعطى سلفاً PRINT LAYOUT SHART و يلاحظ أن يوصف
مكان معين للتاريخ (سنة - شهر - يوم) وكذلك لعدد رقم الصفحات .
وتعطى المتغيرات التي ستدخل مع السطر أسماء متغيرات أو حقول بينما توصيف
باقي السطر فيوصف كـ FILLER

أما جسم التقرير فهو عبارة عن سطر غالباً يكرر على طول الصفحة يحتوي على
حقول متغيرة كل مرة فيوصف سطر واحد هو سطر عمل وتحرك له البيانات ثم يطبع .
أما أسفل الصفحة فقد ينص البرنامج صراحة على طباعة مجاميع في أسفل الصفحة
عندئذ يوصف سطرًا مستقلاً له . وإذا لم يستعمل فيمكن لجسم الصفحة أن يستغل
أسفل الصفحة .

ومن أجل قراءة المخرجات بصورة مريحة وصحيحة يترك فراغ واحد على الأقل بين
سطري طباعة ، و يراعى أن تكون المسافات الأفقية بين الحقول كافية .
وغالباً ما تعالج طباعة العنوان في عنوان فقرة مستقلة تتكرر عندما تكون قيمة عداد
الأسطر مساوية لقيمة معينة ينص عليها البرنامج صراحة .
ومن الأخطاء الشائعة في برامج الطباعة أن المبرمج ينسى أن يحرك القيمة صفراً إلى
عداد الأسطر وبالتالي يتكرر عنده العنوان أكثر من مرة بشكل خاطيء ، لذا يجب
تحريك القيمة صفراً للعداد كلما قلبننا صفحة .

١٣-٢-١ نص التمرين :

اكتب برنامجاً بلغة كوبول يقرأ ملف المشتركين SUBFILE و يطبع محتواه على
الطابعة علماً بأن الملف موجود على قرص ممغنط .

مع مراعاة ما يلي :

- ١ - اطبع العناوين والتاريخ ورقم الصفحة في رأس كل صفحة.
- ٢ - لا تطبع في الصفحة الواحدة أكثر من ٢٥ سطر - عدا العناوين.
- ٣ - اترك سطرًا فارغًا بين كل سطري طباعة.

• في الصفحة التالية يوجد توصيف الطابعة :

| توصيفة | اسم الحقل |
|--------|-----------|
| 9 (5) | D-NO |
| X (20) | D-NAME |
| X (24) | D-ADDRS |
| 9 (8) | D-DATE |
| 99 | D-CODE |
| 99 | D-AREA |
| 9 (7) | D-CNTER |
| 9 (7) | D-LREAD |
| X (5) | D-FILLER |

أنظر الصفحة رقم ٦٠٧ لتوصيف شكل المخرجات والصفحة ٥٩٧ للنتائج.

```

*          حل تمرين طباعه تقرير ١٣-٢-١
IDENTIFICATION DIVISION.
PROGRAM-ID. TXT06.
AUTHOR.      KHAYAT-JNAID.
*
*   يرى هذا التمرين الخطوات الاساسيه في كتابه
*   برنامج يطبع تقرير :
*
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.  IBM-370.
OBJECT-COMPUTER.  IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
        SELECT SUBFILE ASSIGN TO UT-S-SFILE.
        SELECT P-FILE ASSIGN TO UT-S-PRFL.
DATA DIVISION.
FILE SECTION.
*   توصيف ملف الادخال :
*
FD      SUBFILE LABEL RECORD IS STANDARD
01      BLOCK CONTAINS 0 RECORDS DATA RECORD IS SUB-REC
        SUB-REC.
        03      D-NO      PIC 9(5).
        03      D-NAME    PIC X(20).
        03      D-ADDRS   PIC X(24).
        03      D-DATE    PIC X(8).
        03      D-CODE    PIC 99.
        03      D-AREA    PIC 99.
        03      D-CNTR    PIC 9(7).
        03      D-LREAD   PIC 9(7).
        03      FILLER    PIC X(5).
*
*   توصيف ملف الطباعه :
*
FD      P-FILE LABEL RECORD IS OMITTED
        DATA RECORD IS P-REC.
*
*   يوصف ثالبا سجل بطول ١٣ حرفا وهو عرضي الطباعه
*   وتترك السجلات الاخرى الموصوفه في
*   الـ WORKING-STORAGE
*   اليه وتتم الكتابه باسم هذا السجل P-REC ومايحتوى
*   من بيانات مختلفه .
*

```

```

01 P-REC PIC X(132).
*
WORKING-STORAGE SECTION.

```

```

*-----*
* توصف العنوانين كما هو واضح في ورقه *
* التوضيح المرفقه PRINT CHART وتسمى *
* العنوانين الشائنه او الفقرات *
* FILLER بينهما المقول المتغيره *
* مثل التاريخ ورقم الصفحه باسماء *
* مناسبه تذل على مستوى المدخل لتسهيل *
* معالجته *
*-----*

```

```

01 HDR-1.
03 FILLER PIC X(58) VALUE SPACES.
03 FILLER PIC X(16) VALUE
'ELECTRIC COMPANY'.
03 FILLER PIC X(58) VALUE SPACES.
*
01 HDR-2.
03 FILLER PIC X(08) VALUE 'DATE:'.
03 DAY-P PIC 99.
03 FILLER PIC X VALUE '/'.
03 MONTH-P PIC 99.
03 FILLER PIC XXX VALUE '/19'.
03 YEAR-P PIC 99.
03 FILLER PIC X(35) VALUE SPACES.
03 FILLER PIC X(26) VALUE
'LIST OF SUBSCRIBER RECORDS'.
03 FILLER PIC X(39) VALUE SPACES.
03 FILLER PIC X(10) VALUE
' PAGE NO.:'.
03 P-CTR PIC ZZ9.
03 FILLER PIC X(10) VALUE SPACES.

```

```

01 HDR-3.
03 FILLER PIC X(17) VALUE
' SUBSCRIBER NO.'.
03 FILLER PIC X(13) VALUE
' SUBS. NAME'.

```

```

03 FILLER PIC X(20) VALUE SPACES.
03 FILLER PIC X(13) VALUE
  'SUBS. ADDRESS'.
03 FILLER PIC X(11) VALUE SPACES.
03 FILLER PIC X(17) VALUE
  'SUBSCRIPTION DATE'.
03 FILLER PIC X(13) VALUE
  ' SUBS. CODE'.
03 FILLER PIC X(14) VALUE
  ' SUBS. AREA '.
03 FILLER PIC X(14) VALUE
  'COUNTR LAST RD'.

```

```

*-----*
* السطر التالي هو لتوضيح سطر جسم التفسير *
* BODY LINE أي السطر الذي يحتوي على *
* البيانات المتغيرة والتي سيتم تكرارها على طول *
* الصفحة *
*-----*

```

```

01 DATA-LINE.
03 FILLER PIC XX VALUE SPACES.
03 S-NO PIC X(5).
03 FILLER PIC X(08) VALUE SPACES.
03 S-NAME PIC X(20).
03 FILLER PIC X(10) VALUE SPACES.
03 S-ADDRS PIC X(24).
03 FILLER PIC X(10) VALUE SPACES.
03 S-DATE PIC 99/99/9999.
03 FILLER PIC X(08) VALUE SPACES.
03 S-CODE PIC X(2).
03 FILLER PIC X(12) VALUE SPACES.
03 S-AREA PIC X(2).
03 FILLER PIC X(4) VALUE SPACES.
03 S-CNTR PIC Z(5)99.
03 FILLER PIC X(02) VALUE SPACES.
03 S-LREAD PIC Z(5)99.
03 FILLER PIC X VALUE SPACES.

```

```

*
* لفظول التاريخ من الحاسب

```



```

01  CURENT-DATE.
03  C-YEAR  PIC  99.
03  C-MONTH PIC  99.
03  C-DAY   PIC  99.

```

```

*
*      عدادات تستخدم لتجميع
*      عدد الاسطر وعدد الصفحات
77  LN-CTR  PIC  99      VALUE 0.
77  PG-CTR  PIC  999     VALUE 0.

```

```

*
PROCEDURE DIVISION.

```

```

*
*      الخطوة الاولى هي فتح الملفات
OPEN      INPUT  SUBFILE
          OUTPUT P-FILE.

```

```

*
*      قبول التاريخ من الحاسب وتمريكه
*      الى حقول الطباعة ثم طباعه
*      العناوين في الفقرة NEW-PAGE.
*

```

```

ACCEPT  CURENT-DATE FROM DATE.
MOVE    C-YEAR      TO  YEAR-P.
MOVE    C-MONTH     TO  MONTH-P.
MOVE    C-DAY       TO  DAY-P.
PERFORM NEW-PAGE.

```

```

*
MAIN-LOOP.
READ    SUBFILE AT END
        CLOSE SUBFILE P-FILE STOP RUN.

```

```

*
*
*      SUBFILE الإدخال من ملف
*      DATA-LINE الممzروف الى حقول سجل الطباعة
*      الموصوف في قسم توصيف البيانات الإدخالية
MOVE    D-NO      TO  S-NO.
MOVE    D-NAME    TO  S-NAME.
MOVE    D-ADDRS   TO  S-ADDRS.
MOVE    D-DATE    TO  S-DATE.
MOVE    D-CODE    TO  S-CODE.
MOVE    D-AREA    TO  S-AREA.
MOVE    D-CNTR    TO  S-CNTR.
MOVE    D-LREAD   TO  S-LREAD.

```

```

*
*
*   كتابه السجل على ملف الطباعة :
*
      WRITE P-REC FROM DATA-LINE
      AFTER ADVANCING 2 LINES.
*
*
*   إضافة \ لعدد الأسطر LN-C ثم مقارنته
*   محتواه مع العدد 0 لأن عدد الأسطر
*   المسموح طباعتها في الصفحة الواحدة
*   هو 0، فإذا كان أكبر فقلب الصفحة
*   وتطبع العناوين على الصفحة الجديدة
*   ثم يذهب التنفيذ إلى قراءه سجل جديد،
*   وإذا لم يكن أكبر فيذهب التنفيذ إلى
*   معالجه سجل جديد مباشرة.
*
      ADD      1      TO      LN-CTR.
      IF      LN-CTR  >      25
      PERFORM  NEW-PAGE.
      GO TO   MAIN-LOOP.
*
*
*   فخره قلب الصفحة وكتابه العناوين:
*   ( لاحظ إضافة \ لعدد الصفحات C-CNTR )
*   ونحريك القيمة صفر لعدد الأسطر )
      NEW-PAGE.
      MOVE     ZERO      TO      LN-CTR.
      ADD      1      TO      PG-CTR.
      MOVE     PG-CTR    TO      P-CTR.
      MOVE     SPACES    TO      P-REC.
      WRITE    P-REC      AFTER  PAGE.
      WRITE    P-REC FROM HDR-1 AFTER 1 LINE.
      WRITE    P-REC FROM HDR-2 AFTER 1 LINE.
      WRITE    P-REC FROM HDR-3 AFTER 2 LINE.
      EXTP.
      EXIT.

```

١٣ - ٣ برنامج معالجة بيانات عن طريق الأسلوب المباشر:

البرنامج التالي يبين انشاء ملف مفهرس عن طريق استخدام الشاشات TERMINALS وتستخدم غالباً في الأجهزة الصغيرة مثل جهاز تكساس انسترومنتس، بينما الأجهزة الكبيرة مثل IBM فان العمل عليها يمر غالباً باستخدام طرق المعالجة غير المباشرة BATCH MODE لأن استخدام الشاشات TERMINALS يحدد باستخدام برامج ونظم خاصة مثل نظام 'CICS' في جهاز IBM وهو بعيد قليلاً عن لغة كوبول.

والشال التالي يوضح كيفية كتابة برنامج ينشئ ملف مفهرس باستخدام الشاشات. إذ توصف الشاشات ضمن برنامج كوبول باستخدام تعليمة DISPLAY ومعاييرها المختلفة لظهور الرسائل في السطر المعين والموقع المحدد الذي يكون موجوداً أصلاً على ورقة توصيف خاصة بالشاشات.

يحدد المبرمج الشاشات المختلفة التي يريدها. فيبني شاشة الاختيارات الأولى وتسمى عادة MAIN MENU وفيها العمليات الأساسية التي ستجري في البرنامج وغالباً ما تكون: إضافة سجل، تعديل سجل، حذف سجل، استفسار...

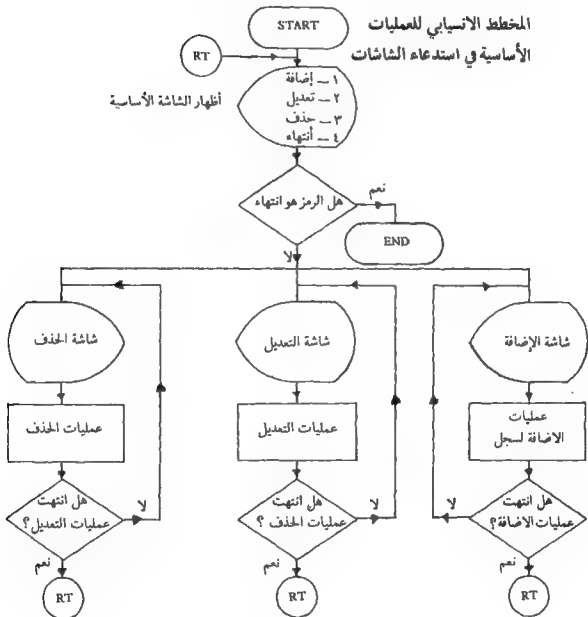
ومن هذه الشاشة تدخل رمز العملية المطلوبة لينتقل التنفيذ منها إلى الشاشات أخرى خاصة بهذا الخيار.

وغالباً ما تكون الشاشات موضح فيها كل العمليات الأساسية التي سيقوم بها مشغل النظام ومن النصائح الأساسية التي يجب مراعاتها عند تصميم الشاشات أن تظهر الرسائل المناسبة عند الضرورة والآن يترك المشغل حائراً ماذا سيفعل ؟ إذ يجب أن يظهر المبرمج رسائل محددة تدل المشغل على الخطوة التالية. مثل (السجل غير موجود، حاول ثانية) الخ... من الرسائل التي تغني جزئياً عن دليل التشغيل

USER GUIDE

ويمكن أن يتطلب عمل تحقق للحقول الداخلة عندها عند قبول كل حقل يعمل تحقق عليه وتظهر الرسائل المناسبة في حال الخطأ أمام المدخل بصورة خاطئة.

و ينتهى البرنامج غالباً برمز معين يدخل ضمن الشاشة الرئيسية وليس هناك شرط
 لنهاية ملفات أو غيره غالباً .
 الشكل التالي يظهر تسلسل استدعاء الشاشات ضمن البرنامج .



١٣-٣-١ نص التمرين :

المطلوب كتابة برنامج بلغة الكوبول ينشيء (CREATE) الملف الأساسي للمشاركون واسمه SUBSFL على أن يكون هذا الملف من نوع المفهرس INDEX SEQUENTIAL وذلك عن طريق الشاشات TERMINALS (وفق النموذج المرفق — حالة TI) .

كما و يطلب أن يكون البرنامج يسمح بعمليات التحديث UPDATE على الملف ، والتي تشمل تعديل سجل CHANGE أو إضافة INSERT أو حذف سجل DELETE وتتم العمليات وفق ما يلي :

أ — مرحلة إضافة سجلات جديدة INSERT NEW RECORDS :

يقبل عن طريق الشاشة TERMINAL المعلومات التالية :

- ١ — رقم المشترك SUBS-NO يجب أن يكون رقمياً وأكبر من الصفر.
- ٢ — اسم المشترك SUBS-NAME يجب ألا يكون رقمياً أو فراغ.
- ٣ — عنوان المشترك SUBS-ADRES يجب ألا يكون فراغ.
- ٤ — تاريخ الاشتراك SUBS-DATE :
- SUBS-DAY اليوم يجب أن يكون محصوراً بين ١ — ٣٠ .
- SUB-MONTH الشهر يجب أن يكون محصوراً بين ١ — ١٢ .
- SUBS-YEAR السنة يجب أن تكون بين ١٤٠١هـ — ١٤١٠هـ .
- ٥ — رمز الاشتراك SUBS-CODE يجب أن يكون رقمياً ومحصوراً بين ١ — ١٠ .
- ٦ — رمز المنطقة SUBS-AREA يجب أن يكون رقمياً ومحصوراً بين ١ — ٢٠ .
- ٧ — رقم العدد SUBS-CNTR يجب أن يكون رقمياً وأكبر من الصفر.
- ٨ — مقدار القراءة SUBS-LREAD يجب أن يكون رقمياً وأكبر من الصفر.

عند قبول الحقول السابقة يجب عمل تحقق VALIDATION حسب القواعد المذكورة سابقاً، وعند وجود أي حقل خطأ أظهر عبارة 'ERROR' أمام الحقل الخطأ، وكرر قبول الحقل من جديد. وعند تكرار الخطأ ثلاثة مرات للحقل الواحد أرفض السجل وأظهر رسالة 'REJECTED RECORD' وأذهب لتقبل سجل جديد.

أما عند خلو الحقول من الأخطاء أكتب هذا السجل على الملف SUBSFL وفي حال نجاح الكتابة أظهر رسالة 'SUCCESSFUL INSERTION' وكرر العمل من أجل سجل جديد، أما في حالة الخطأ INVALID KEY CONDITION فأظهر رسالة 'DUPLICATE KEY' وكرر العمل من أجل سجل جديد.

في نهاية الاضافة ادخل الرمز المناسب للخروج من حالة الاضافة والذهاب إلى الشاشة الرئيسية MAIN MENU في البرنامج.

ب — تعديل سجل UPDATE :

و يطلب في هذه المرحلة قبول رقم المشترك المراد التعديل على بياناته .
ثم يقرأ سجل المشترك وفق الرقم المعطى عن طريق الشاشة .
في حال كون القراءة خاطئة أظهر عبارة 'RECORD NOT EXIST' وأذهب لتقبل رقم جديد.

أما في حال كون القراءة ناجحة فأظهر محتوى هذا السجل المقروء على الشاشة وفق النموذج المرفق وأظهر في أسفل الشاشة عبارة :

'ENTER THE FIELD NUMBER :'

أي ادخل رقم الحقل المراد التعديل عليه : أقبل رقم الحقل وعدّل الحقل المرافق .
ولا تنسى عمل تحقق VALIDATION له . وفي النهاية أدخل الرمز المناسب لالتهاء من تعديل هذا السجل أو أعد كتابة السجل على الملف .

كرر العمل من أجل سجل آخر، وفي نهاية عمليات التعديل أدخل الرمز المناسب للعودة للشاشة الرئيسية في البرنامج.

جـ - عملية حذف سجل DELETE A RECORD :

١ - اقبل عن طريق الشاشة رقم المشترك المراد حذف سجله .

٢ - اقرأ السجل بالمفتاح السابق .

وفي حال القراءة خاطئة أظهر رسالة 'RECORD NOT EXIST' السجل غير موجود وأذهب للخطوة رقم ١ .

أما إذا كان السجل موجود فأظهره على الشاشة وأظهر رسالة:

'ARE YOU SHURE ? Y/N' هل أنت متأكد ؟ ن/لا

فإذا كانت الاجابة لا أي أنه ليس هذا السجل المطلوب فارجع للخطوة رقم ١ .

والآ احذف هذا السجل وأذهب للخطوة رقم ١ .

٣ - في نهاية عمليات الحذف ادخل الرمز المناسب للخروج من عمليات الحذف

MAIN MENUE الشاشة الرئيسية

ادخل الرمز المناسب في الشاشة الرئيسية للخروج من البرنامج.

توصيف الشاشة الأساسية :

SELECT ONE OPTION :

1. INSERT NEW RECORD.
2. UPDATE EXIST RECORD.
3. DELETE EXIST RECORD.
4. EXIT

FILE NAME PIC : SUBSFL توصيف الملف

| | | |
|-----------------|-------------|--------|
| رقم المشترك | SUBS-NO | X (5) |
| اسم المشترك | SUBS-NAME | X (30) |
| عنوانه | SUBS-ADRESS | X (24) |
| تاريخ الاشتراك | SUBS-DATE | |
| يوم | SUBS-DAY | 99 |
| شهر | SUBS-MONTH | 99 |
| سنة | SUBS-YEAR | 9 (4) |
| رمز الاشتراك | SUBS-CODE | 99 |
| رمز المنطقة | SUBS-AREA | 99 |
| رقم العداد | SUBS-CNTR | 9 (7) |
| مقدار آخر قراءة | SUBS-LREAD | 9 (7) |
| فراغ | SUBS-FLR | X (4) |

٢ - ٣ - ١٣

نص برنامج معالجة بيانات :

```

IDENTIFICATION DIVISION.
PROGRAM-ID.      ONLNUPD.
AUTHOR.          KHAYAT-JNAID.

* ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. T1990.
OBJECT-COMPUTER. T1990.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT SUBSFL ASSIGN TO RANDOM "SM"
    ORGANIZATION IS INDEXED;
    ACCESS MODE IS RANDOM;
    RECORD KEY IS SUBS-NO.

* DATA DIVISION.
FILE SECTION.

```


FD SUBSFL LABEL RECORD STANDARD
RECORD CONTAINS 80 CHARACTERS.

01 SUBS-REC.
03 SUBS-NO PIC X(05).
03 SUBS-NAME PIC X(20).
03 SUBS-ADRES PIC X(24).
03 SUBS-DATE.
05 SUBS-DAY PIC 99.
88 DAYY VALUES ARE 01 THRU 30.
05 SUBS-MONTH PIC 99.
88 MONTHH VALUES ARE 01 THRU 12.
05 SUBS-YEAR PIC 9999.
88 YEARR VALUES ARE 1401 THRU 1410.
03 SUBS-CODE PIC 99.
88 CODEE VALUES ARE 01 THRU 10.
03 SUBS-AREA PIC 99.
88 AREAA VALUES ARE 01 THRU 20.
03 SUBS-CNTR PIC 9(7).
03 SUBS-LREAD PIC 9(7).
03 FILLER PIC X(5).

* WORKING-STORAGE SECTION.

* 01 DEC-CODE PIC 99
USAGE IS COMP-1.
01 BIN-VAL REDEFINES DEC-CODE PIC XX.
01 CHAR PIC 99.
77 FLAG PIC 9 VALUE ZERO.
77 FLD-NO PIC 99.
77 DUMMY PIC XX.
77 ANSR PIC X.
77 OPTIONS PIC 9.

* PROCEDURE DIVISION.
STRT.

OPEN I-O SUBSFL.
* انشغاليات الفلاش المتتالية تستخدم لتحويل
* نمط الكتابة على الشاشة إلى العشري.

* MOVE 18 TO DEC-CODE.
MOVE BIN-VAL TO CHAR.
DISPLAY CHAR.

* MENU.
DISPLAY

"برنامج بناء وتعديل الملفات JCL"
LINE 3 POSITION 24 ERASE
"الشاشة الكهربية"
LINE 4 POSITION 30
"-----"
LINE 5 POSITION 24
"إضافة سجل جديد"
LINE 6 POSITION 27
"تعديل سجل موجود 2-"

LINE 8 POSITION 27
 "خط سفل موجود 3-"
 LINE 10 POSITION 27
 "لاستفسار عن سجل 4-"
 LINE 12 POSITION 27
 "إدخال رقم السجل أو رقم RETURN لا يتجاوز"
 LINE 14 POSITION 20
 "-----"

LINE 16 POSITION 20.
 ACPT-COD.
 ACCEPT OPTIONS
 LINE 14 POSITION 70 PROMPT TAB.
 IF OPTIONS LESS THAN 1 OR
 OPTIONS GREATER THAN 4
 CLOSE SUBSFL STOP RUN.

* قبول رقم السجل من الشاشة
 * ومعرفة هل نحن امام شاشة ام خط ام تعديل
 *
 MOVE HIGH-VALUE TO SUBS-NO.
 IF OPTIONS = 1 PERFORM INSRT-RTN THRU EXT-INS
 UNTIL SUBS-NO = "00000"
 IF OPTIONS = 2 PERFORM UPDAT-RTN THRU EXT-UPD
 UNTIL SUBS-NO = "00000"
 IF OPTIONS = 3 PERFORM DELT-RTN THRU EXT-DÉL
 UNTIL SUBS-NO = "00000"
 IF OPTIONS = 4 PERFORM INGR-RTN THRU EXT-ING
 UNTIL SUBS-NO = "00000"
 GO TO MENU.

* المفرد الخاصة بالتعديل ولها قسم قدره السجل المطلوب
 * يتم تظهر معلومات على الشاشة وتعديل ختلفة حسب المطلوب
 * REWRITE وتعاد كتابة السجل على الملف بتعليقه
 *
 UPDAT-RTN.
 PERFORM ACCEPT-NO THRU EXT-AC.
 IF SUBS-NO = "00000" GO TO EXT-UPD.
 IF FLAG = 1
 DISPLAY "السمجل غير موجود"
 LINE 20 POSITION 10
 ACCEPT DUMMY GO TO EXT-UPD.
 PERFORM DISPLAY-SCREEN THRU EXT-SC.
 PERFORM DISPLAY-FIELDS THRU EXT-FLD.
 DISPLAY
 "إدخال رقم السجل أو الرقم صفر لا يتجاوز"
 LINE 23 POSITION 5.
 PERFORM ACCEPT-FLD THRU EXT-UPD
 UNTIL FLD-NO = ZERO.
 REWRITE SUBS-REC.
 DISPLAY
 "السمجل-----"
 LINE 23 POSITION 5.

```

ACCEPT DUMMY.
GO TO UPDAT-RTN.
ACCEPT-FLD.
ACCEPT FLD-NO LINE 23 POSITION 75 PROMPT TAB.
IF FLD-NO = 2 PERFORM AC-NAME ELSE
IF FLD-NO = 3 PERFORM AC-ADRES ELSE
IF FLD-NO = 5 PERFORM AC-YEAR ELSE
IF FLD-NO = 6 PERFORM AC-MONTH ELSE
IF FLD-NO = 7 PERFORM AC-DAY ELSE
IF FLD-NO = 8 PERFORM AC-CODE ELSE
IF FLD-NO = 9 PERFORM AC-AREA ELSE
IF FLD-NO = 10 PERFORM AC-CNTR ELSE
IF FLD-NO = 11 PERFORM AC-LREAD.
EXT-UPD.
EXIT.

```

*
* الفخريه الشالبيه هي لاضافه سجل جديد وعمل تحقق عليه
* ثم الشالبيه على السجل بشالبيهه
*
* WRITE

```

INSRT-RTN.
PERFORM ACCEPT-NO THRU EXT-AC.
IF SUBS-NO = "00000" GO TO EXT-INS.
IF FLAG = ZERO DISPLAY
"السجل مكرر من السجل الرابع اعد السجل"
LINE 20 POSITION 10
ACCEPT DUMMY GO TO EXT-INS.
PERFORM DISPLAY-SCREEN THRU EXT-SC.
DISPLAY SUBS-NO LINE 3 POSITION 45.

```

*
* من الفخريه الشالبيه يتم قبول كل سجل ويكمل تحقق عليه ونظهر
* رساله خطأ في حال حدوثه بعدد قبوله من جديد
* *** لاحظ استخدام اسم الفخريه عند التحقق من دخول المتابعه من
* AC-NAME.

```

ACCEPT SUBS-NAME LINE 5 POSITION 45 PROMPT TAB.
IF SUBS-NAME = SPACES OR SUBS-NAME NUMERIC
GO TO AC-NAME.

```

```

AC-ADRES.
ACCEPT SUBS-ADRES LINE 6 POSITION 45 PROMPT TAB.
IF SUBS-ADRES = SPACES GO TO AC-ADRES.

```

```

AC-YEAR.
ACCEPT SUBS-YEAR LINE 08 POSITION 45 SIZE 4
PROMPT TAB.
IF NOT YEARR GO TO AC-YEAR.

```

```

AC-MONTH.
ACCEPT SUBS-MONTH LINE 09 POSITION 45 SIZE 2
PROMPT TAB.
IF NOT MONTHH GO TO AC-MONTH.

```

```

AC-DAY.
ACCEPT SUBS-DAY LINE 10 POSITION 45 SIZE 2
PROMPT TAB.
IF NOT DAY GO TO AC-DAY.

```

```

AC-CODE.
ACCEPT SUBS-CODE LINE 11 POSITION 45 PROMPT TAB.
IF NOT CODEE GO TO AC-CODE.

```

```

AC-AREA.
ACCEPT SUBS-AREA LINE 12 POSITION 45 PROMPT TAB.

```

```

IF NOT AREA0 GO TO AC-AREA.
*
AC-CNTR.
ACCEPT SUBS-CNTR LINE 13 POSITION 45 PROMPT TAB.
IF SUBS-CNTR NOT NUMERIC OR SUBS-CNTR = ZERO
GO TO AC-CNTR.
*
AC-LREAD.
ACCEPT SUBS-LREAD LINE 14 POSITION 45 PROMPT TAB.
IF SUBS-LREAD NOT NUMERIC OR SUBS-LREAD = ZERO
GO TO AC-LREAD.
*
كتابة المسجل على الشريط بعد ان شاكذنا من انه صحيح
*
WRITE SUBS-REC
INVALID KEY DISPLAY "الرجاء اعد التسجيل"
LINE 23 POSITION 10
ACCEPT DUMMY GO TO EXT-INS.
DISPLAY "الرجاء اعد التسجيل"
LINE 23 POSITION 10 ACCEPT DUMMY.
EXT-INS.
EXIT.
*
الشفرة التالية هي للاستفسار عن سجل دون إمكانية تحديثه
*
INQR-RTN.
PERFORM ACCEPT-NO THRU EXT-AC.
IF SUBS-NO = "00000" GO TO EXT-ING.
IF FLAG = 1
DISPLAY "الرجاء اعد التسجيل"
LINE 20 POSITION 10
ACCEPT DUMMY GO TO EXT-UPD.
PERFORM DISPLAY-SCREEN.
PERFORM DISPLAY-FIELDS.
DISPLAY
"الخط يحتاج الى تحديثه على الاستفسار من جديد"
LINE 23 POSITION 5 ACCEPT DUMMY.
GO TO INQR-RTN.
EXT-ING.
EXIT.
*
*
الشفرة التالية هي لحذف سجل من الملفات بتعليقه
*
DELETE
*
DELT-RTN.
PERFORM ACCEPT-NO THRU EXT-AC.
IF SUBS-NO = "00000" GO TO EXT-DEL.
IF FLAG = 1 DISPLAY
"ليس هناك سجل لحذفه حتى شاكذ من الرجاء"
LINE 20 POSITION 10
ACCEPT DUMMY GO TO EXT-DEL.
DISPLAY SUBS-NAME LINE 20 POSITION 10 ERASE
"هل اريد حذفه او لا"
LINE 22 POSITION 5.
ACCEPT ANSR LINE 21 POSITION 10 PROMPT TAB.
IF ANSR = "ع" GO TO EXT-DEL.
*
[ اذا كانت الاجابة ن اى نعم يتم الحذف كما هي لتعليقه التالية
*
DELETE SUBSFL RECORD.
DISPLAY "تم حذف المسجل" LINE 23 POSITION 30.
ACCEPT DUMMY.
EXT-DEL.
EXIT.

```


١٣ - ٤ برنامج معالجة بيانات وطباعة :

البرنامج التالي يعالج ملفين أحدهما تسلسلي SEQUENTIAL والآخر مفهرس

INDEXED SEQUENTIAL

يقرأ الملف التسلسلي و يؤخذ المفتاح منه لقراءة الملف المفهرس ، وعندما تكون القراءة صحيحة VALID KEY تتم معالجة المطلوب وهو مثلاً حساب فاتورة المستهلك باستعمال حقول من الملفين وطباعة الناتج على الطابعة . ثم معاودة القراءة من جديد . هذا الأسلوب من المعالجة بسيط إذ يستخدم ثلاثة ملفات الملف التسلسل والملف المفهرس وملف الطابعة لطباعة الناتج .

أحياناً يطلب في البرنامج تعديل حقول ضمن سجل الملف المفهرس لذلك يحرك الحقل المرغوب (في مثالنا هنا مقدار القراءة الحالية إلى مقدار القراءة الأخيرة) من ملف الادمخال إلى الملف المفهرس و يعاد كتابة السجل REWRITE وذلك لتعديل حقول أثناء المعالجة .

انظر البرنامج :

١٣ - ٤ - ١ نص التمرين :

لديك ملفان : الأول واسمه INDEXFL وهو ملف مفهرس يحتوي على معلومات عن المشتركين في شركة الكهرباء وشكل سجله :

| | FIELD NAME | PICTURE |
|----------------|------------|---------|
| رقم المشترك | IDX-NO | X (5) |
| أسمه | IDX-NAME | X (20) |
| عنوانه | IDX-ADRESS | X (24) |
| تاريخ الاشتراك | IDX-DATE | 9 (8) |
| رمز الاشتراك | IDX-CODE | 99 |

| | | |
|-----------------|-----------|-------|
| رمز المنطقة | IDX-AREA | 99 |
| رقم العدد | IDX-CNTR | 9 (7) |
| القراءة الأخيرة | SDX-LREAD | 9 (7) |
| فراغ | BLANK | X (5) |

والثاني واسمه NCONSMF وهو ملف متسلسل يحتوي على معلومات عن استهلاك المشتركين في شركة الكهرباء وشكل سجله :

| FIELD NAME | PICTURE |
|-----------------------|-------------------|
| رقم المشترك | CONS-NO 9 (5) |
| مقدار القراءة الحالية | CONS-READ 9 (7) |
| تاريخ القراءة | CONS-DATE 9 (8) |
| ملاحظات | CONS-COMNT X (30) |
| فراغ | FILLER X (30) |

المطلوب اقرأ ملف المستهلكين NCONSMF أولاً ثم حرك الحقل CONS-NO إلى مفتاح الملف INDXFL وهو IDX-NO وقرأ ملف المشتركين الأساسي بالمفتاح السابق وإذا لم يوجد هذا السجل أعد قراءة سجل جديد من ملف المستهلكين وإلا تابع العمل، كالتالي :

المطلوب طباعة قائمة بالمشاركين واستهلاكهم والمستحق عليهم علماً بأن الاستهلاك يحسب بالطريقة التالية :

- ١ - يطرح مقدراً القراءة السابقة IDX-LREAD من مقدار القراءة الحالية CONS-READ ويوضح الناتج في حقل اسمه W-READ وهو يمثل الاستهلاك.
- ٢ - إذا كان رمز الاشتراك ١ أو كان الاستهلاك W-READ أقل من ١٠٠ كيلوات ساعي فيحسب سعر الكيلوات بـ ٥ هملات.
- ٣ - إذا كان رمز الاشتراك ٢ أو كان الاستهلاك W-READ أكبر من ٩٩ كيلوات

- ساعي وأصغر من ٥٠٠ فيحسب سعر الكيلووات ساعي بـ ٤ هللات .
- ٤ - إذا كان رمز الاشتراك ٣ أو كان الاستهلاك W-READ أكبر من ٥٠٠ كيلووات ساعي فيحسب سعر الكيلووات بـ ٣ هللات .
- ٥ - إذا كان رمز الاشتراك من : ٠٤ إلى ٠٦ فيعفى من دفع قيمة الاستهلاك .
- ٦ - إذا كان رمز الاشتراك من ٠٧ - ١٠ فيحسب سعر الكيلووات ساعي بـ ٢ هللة .
- ٧ - تضاف قيمة رسوم الجباية بمعدل ٥٠ هللة لكل مشترك مهما كان رمز اشتراكه أو مقدار استهلاكه .
- ٨ - تضاف قيمة ٢% من قيمة الفاتورة الاجالية كرسوم تحصيل .
- ٩ - تجمع كل القيم السابقة في حقل يسمى المستحق للدفع .
- ١٠ - ينقل مقدار القراءة الحالية CONS-READ إلى حقل IDX-READ الموجود في ملف المشتركين الأساسي ويعاد كتابة السجل في الملف الأساسي .
- ١١ - يطبع السجل كما هو وارد في التوصيف المرفق .
- ١٢ - يكرر العمل حتى نهاية ملف الاستهلاك .
- ١٣ - يطبع في نهاية العمل : عدد المستهلكين : عدد الكيلووات المصروفة : مجموع المبالغ المستحقة .

ملاحظات :

- ١ - اطبع التاريخ ورقم الصفحة والعناوين برأس كل صفحة .
- ٢ - اطبع في الصفحة الواحدة ٢٥ سطراً فقط .
- ٣ - اطبع المجاميع على صفحة مستقلة .
- ٤ - في حالة تضارب شرطين في حساب الفاتورة يؤخذ بالحالة الأقل تكلفة بالنسبة للمستهلك . أنظر صفحة رقم ٦٠٨ من أجل توصيف شكل المخرجات والصفحة رقم ٥٩٨ للنتائج

١٣- ٤- ٣ نص برنامج معالجة البيانات والطباعة

• هل تمرين معالجته بيانات وطباعتها ١٣-٤-٣ *

IDENTIFICATION DIVISION.
PROGRAM-ID. TXT18.
AUTHOR. KHAYAT-JNAID.

*
* يرى هذا التمرين الخطوات الأساسية في كتابه
* برنامج يعالج البيانات ثم يطبع النتائج.
*

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT IDXFL ASSIGN TO IDXFL
ORGANIZATION IS INDEXED
ACCESS MODE IS RANDOM
RECORD KEY IS IDX-NO.
SELECT NCONSP ASSIGN TO UT-S-NCON.
SELECT HGRIDAT ASSIGN TO UT-S-HDAT.
SELECT P-FILE ASSIGN TO UT-S-PRFL.

DATA DIVISION.
FILE SECTION.

* توصيف ملفات الإدخال : *

* . INDXFL وهو الملف الأساسي المفهرس
* والمطلوب معالجته وتعديله.
*

FD INDXFL LABEL RECORD IS STANDARD
DATA RECORD IS IDX-REC.

01 IDX-REC.
03 IDX-NO PIC X(5).
03 IDX-NAME PIC X(20).
03 IDX-ADRS PIC X(24).
03 IDX-DATE PIC 9(8).
03 IDX-CODE PIC 99.
03 IDX-AREA PIC 99.
03 IDX-CNTR PIC 9(7).
03 IDX-LREAD PIC 9(7).
03 FILLER PIC X(5).

*
 * NCONSMF وهو ملف الحركة والذي يحتوى على
 * المعلومات الجديدة المعدلة (بكر الدال)
 *

FD NCONSMF LABEL RECORD IS STANDARD
 BLOCK CONTAINS 0 RECORDS
 DATA RECORD IS CON-REC.

01 CON-REC.
 03 CON-NO PIC X(5).
 03 CON-READ PIC 9(7).
 03 CON-DATE.
 05 CON-YEAR PIC 9999.
 05 CON-MONT PIC 99.
 05 CON-DAY PIC 99.
 03 CON-CMNT PIC X(30).
 03 FILLER PIC X(30).

*
 * يحتوى على التاريخ العجى لمصوبه
 * الحصول عليه من القاسم

FD HGRIDAT LABEL RECORD IS OMITTED
 BLOCK CONTAINS 0 RECORDS
 DATA RECORD IS H-REC.

01 H-REC.
 05 H-MONTH PIC 99.
 05 H-YEAR PIC 9999.
 05 FILLER PIC X(74).

*
 * توصيف ملف الطباعة :
 *

FD P-FILE LABEL RECORD IS OMITTED
 BLOCK CONTAINS 0 RECORDS.

*
 * يوم غاليا سجل بطول ١٣ حرفا وهو عرضى الطباعة
 * وتترك السجلات الاخرى الموضحة فى WORKING-STORAGE
 * اليه وتتم الكتابة باسم هذا السجل P-REC وما يحتوى
 * من بيانات مختلفة .
 *

01 P-REC PIC X(132).

*
 WORKING-STORAGE SECTION.

```

*-----*
* توصف العناوين كما هو واضح في ورقه *
* التوضيح المرفقه PRINT CHART وتسمى *
* العناوين الثابته او الخزائن *
* ب FILLER بينما الدقول المتغيره *
* مثل التاريخ ورقم الصفحه باسماء *
* مناسبه تدل على مستوى الدقيل لتسهيل *
* معالجه *
*-----*
01 HDR-1.
03 FILLER PIC X(58) VALUE SPACES.
03 FILLER PIC X(16) VALUE
  'شركه الكوربا'
03 FILLER PIC X(58) VALUE SPACES.
*
01 HDR-2.
03 FILLER PIC XX VALUE '19'.
03 YEAR-P PIC 99.
03 FILLER PIC X VALUE '/'.
03 MONTH-P PIC 99.
03 FILLER PIC X VALUE '/'.
03 DAY-P PIC 99.
03 FILLER PIC X(08) VALUE 'تاريخ'.
03 FILLER PIC X(35) VALUE SPACES.
03 FILLER PIC X(26) VALUE
  'قامه بالمشاركين واستقلاهم'.
03 FILLER PIC X(39) VALUE SPACES.
03 P-CTR PIC ZZ9.
03 FILLER PIC X(10) VALUE
  'صفحه رقم:'.
03 FILLER PIC X(10) VALUE SPACES.
*
01 HDR-3.
03 FILLER PIC X(60) VALUE SPACES.
03 HDYEAR PIC 9999.
03 FILLER PIC X(05) VALUE 'عام'.
03 HDMONT PIC 99.
03 FILLER PIC X(62) VALUE 'لشهر'.
*
01 HDR-4.
03 FILLER PIC X(20) VALUE

```

```

      ' المستحق للدفع '
03  FILLER PIC X(22) VALUE
      ' العدد مقدار الاستهلاك '
03  FILLER PIC X(28) VALUE
      ' رمز المنطقة رمز الاستهلاك رقم '
03  FILLER PIC X(20) VALUE
      ' عنوان المشترك '
03  FILLER PIC X(21) VALUE
      ' اسم المشترك '
03  FILLER PIC X(19) VALUE
      ' رقم المشترك '

*-----*
* السطر التالي هو لتوضيح سطر جسم المضيف *
* BODY LINE في السطر الذي يحتوي على *
* البيانات المتغيرة والذي سينكر على طول *
* الصفحة *
*-----*
01  DATA-LINE.
03  FILLER PIC XXX VALUE SPACES.
03  P-PAY PIC Z(6)9.99.
03  FILLER PIC X(11) VALUE SPACES.
03  P-CONSMF PIC Z(5)99.
03  FILLER PIC X(07) VALUE SPACES.
03  P-CONTR PIC Z(5)99.
03  FILLER PIC X(09) VALUE SPACES.
03  P-CODE PIC 99.
03  FILLER PIC X(09) VALUE SPACES.
03  P-AREA PIC 99.
03  FILLER PIC X(05) VALUE SPACES.
03  P-ADRESS PIC X(24).
03  FILLER PIC XX VALUE SPACES.
03  P-NAME PIC X(20).
03  FILLER PIC X(05) VALUE SPACES.
03  P-NO PIC X(5).
03  FILLER PIC X(8) VALUE SPACES.

*
01  FIN-1.
03  FILLER PIC X(58) VALUE SPACES.
03  FILLER PIC X(16) VALUE
      ' الفلوس النهائية '
03  FILLER PIC X(58) VALUE SPACES.

```

```

01  FIN-2.
03  FILLER      PIC  X(50) VALUE SPACES.
03  PCNTR1      PIC  ZZZ999.
03  FILLER      PIC  X(68) VALUE
      'عدد المستهلكين :
01  FIN-3.
03  FILLER      PIC  X(50) VALUE SPACES.
03  PCNTR2      PIC  Z(7)99.
03  FILLER      PIC  X(64) VALUE
      'مقدار الكيلو وات المصروفة :
01  FIN-4.
03  FILLER      PIC  X(50) VALUE SPACES
03  PCNTR3      PIC  Z(7)9.99.
03  FILLER      PIC  X(62) VALUE
      'مقدار المبالغ المستدقة :
*
*
*  مساحة لقبول التاريخ
01  CUR-DATE.
03  C-YEAR      PIC  99.
03  C-MONTH     PIC  99.
03  C-DAY       PIC  99.
*
77  F-ST        PIC  99          VALUE ZERO.
*  أعداد تستخدم لتجميع
*  عدد الأسطر وعدد الصفحات
77  LN-CTR      PIC  99          VALUE 30.
77  P-CNTR      PIC  999        VALUE ZERO.
*  أعداد تستخدم كمساهمات عمل وسيطه
77  W-READ      PIC  9(7).
77  W-PAY       PIC  9(5)V99.
77  HALALAH     PIC  9V99.
*  أعداد تستخدم لتجميع
*  المبالغ أو المقادير المصروفة
77  CNTR1       PIC  9(6)        VALUE ZERO.
77  CNTR2       PIC  9(9)        VALUE ZERO.
77  CNTR3       PIC  9(8)V99    VALUE ZERO.

```

```

*
PROCEDURE DIVISION.
*
* الخطوة الاولى هي فتح الملفات
OPEN          INPUT      NCONSMF HGRIDAT
              I-O        INDXFL
              OUTPUT     P-FILE.
*
* قبول التاريخ من الماسك وتحويله
* الى جدول الطباعة ثم طباعه
* العناوين في الفقره NEW-PAGE
*
ACCEPT CUR-DATE FROM DATE.
PERFORM INSP      THRU EXTS.
MOVE SPACE TO P-REC.
WRITE P-REC AFTER PAGE.
MOVE C-YEAR TO YEAR-P.
MOVE C-MONTH TO MONTH-P.
MOVE C-DAY TO DAY-P.
READ HGRIDAT.
MOVE H-MONTH TO HDMONT.
MOVE H-YEAR TO HDYEAR.
CLOSE HGRIDAT.
PERFORM NEW-PAGE.
*
MAIN-LOOP.
*
* الفقره من ملف التعديل الاولى
READ NCONSMF AT END
PERFORM FINAL-RTN
CLOSE NCONSMF INDXFL
P-FILE STOP RUN.
*
* ثم تحويل رقم المشترك من ملف الاستهلاك
* الى جدول رقم المشترك في الملف الاساسي
* والفقره به عشوائيا RANDOMLY حسب قيمه
* المفتاح فاذا تمت الفقره اي السجل
* موجود فينقل التنفيذ للتعليمة
* PERFORMANCE INVALID KEY
* ولا تتشكل حاله
* التنفيذ الى الفقره MAIN-LOOP للفقره
* من جديد
*
MOVE CON-NO TO IDX-NO.

```

```

READ   INDXFL  INVALID KEY
      GO TO  MAIN-LOOP.
PERFORM CALCULATE-RTN.
GO TO  MAIN-LOOP.

```

```

*
*   الذخيرة التالية هي لحساب استهلاك مشترك
*
CALCULATE-RTN.
      MOVE  ZERO      TO  HALALAH.
*
*   حساب الاستهلاك بالكيلو وات
*
      SUBTRACT INDX-LEAD FROM CON-READ
      GIVING W-READ.
*
*   تمديد المستهلك تبعا لاستهلاكه أو
*   لرمز اشتراكه، ثم وضع قيمه استهلاك
*   الوحدة في حقل HALALAH لحساب مجمل
*   الاستهلاك.
*
      IF  INDX-CODE = 01 OR W-READ < 100
      MOVE 0.05 TO HALALAH.
*
      IF  INDX-CODE = 02 OR
      ( W-READ > 99 AND W-READ < 500 )
      MOVE 0.04 TO HALALAH.
*
      IF  INDX-CODE = 03 OR W-READ > 500
      MOVE 0.03 TO HALALAH.
*
      IF  INDX-CODE > 06 AND INDX-CODE < 11
      MOVE 0.02 TO HALALAH.
*
      IF  INDX-CODE > 05 AND INDX-CODE < 07
      MOVE  ZEROS TO HALALAH.
*
*   حساب التكاليف
*
MULTIPLY W-READ BY HALALAH
      GIVING W-PAY.
ADD 0.50 TO W-PAY.

```

```

*
* حساب ٣٠% من التكلفة وإضافه الناتج الى
* التكلفة مباشرة (أي مضروب ب ١٣٠ % ) .
*
*
*      MULTIPLY 1.20 BY W-PAY.
*
*
* تمريرك حقل آخر ظرفه CON-READ من ملف
* استهلاك الى الملف الأساسي وإعادته كتابه
* السجل بعد تعديله .
*
*      MOVE      CON-READ TO      IDX-LREAD.
*      REWRITE   IDX-REC.
*
*      طباعه السجل على الطباعه
*      PERFORM   PRINT-RTN.
*
*
* تجميع عدد المشتركين والاستهلاك والمبالغ
* المستحقة في عدداتها المناسبه
*
*
*      ADD      1          TO      CNTR1.
*      ADD      W-READ     TO      CNTR2.
*      ADD      W-PAY      TO      CNTR3.
*
*      EXT.
*      EXIT.
*
*
*
* تمريرك المقول من ملفات الإدخال, وكذلك
* المقول الحسابيه الوسيظه الى DATA-LINE
* الموصوف في قسم توصيف البيانات الإدخليه
*
*
*      PRINT-RTN.
*      MOVE     IDX-NO      TO      P-NO.
*      MOVE     IDX-NAME    TO      P-NAME.
*      MOVE     IDX-ADRS    TO      P-ADDRESS.
*      MOVE     IDX-CODE    TO      P-CODE.
*      MOVE     IDX-AREA    TO      P-AREA.
*      MOVE     IDX-CNTR    TO      P-CONTR.
*      MOVE     W-READ      TO      P-CONSUMP.
*      MOVE     W-PAY       TO      P-PAY.

```



```

*
*
* كتابة السجل على ملف الطباعة :
*
      MOVE      DATA-LINE TO P-REC.
      PERFORM   INSP          THRU EXTS.
      WRITE     P-REC
                AFTER ADVANCING 2 LINES.

*
*
* إضافة | لعدد الأسطر LN-CTR ثم مقارنته
* معنواة مع العدد ٢٥ لأن عدد الأسطر
* المسموع طباعتها في الصفحة الواحدة
* هو ٢٥ ، فإذا كان أكبر قلب الصفحة
* وتطبع العناوين على الصفحة الجديدة
* ثم يذهب التنفيذ إلى قراءة سجل جديد ،
* وإذا لم يكن أكبر فيذهب التنفيذ إلى
* معالجة سجل جديد مباشرة .
*
      ADD       1          TO      LN-CTR.
      IF        LN-CTR    >      25
                PERFORM   NEW-PAGE.

EXTL.
EXIT.

*
*
* دفتر قلب الصفحة وكتابه العناوين :
* ( لاحظ إضافة | لعدد الصفحات PG-CTR
* وتمريك القيمة صفر لعدد الأسطر )
NEW-PAGE.
      MOVE      ZERO      TO      LN-CTR.
      ADD       1          TO      P-CNTR.
      MOVE      P-CNTR    TO      P-CTR.
      MOVE      SPACES    TO      P-REC.
      WRITE     P-REC      AFTER   PAGE.
      MOVE      HDR-1     TO      P-REC.
      PERFORM   INSP          THRU EXTS.
      WRITE     P-REC      AFTER 1 LINE.
      MOVE      HDR-2     TO      P-REC.
      PERFORM   INSP          THRU EXTS.
      WRITE     P-REC      AFTER 2 LINE.
      MOVE      HDR-3     TO      P-REC.
      PERFORM   INSP          THRU EXTS.

```

```

WRITE P-REC AFTER 1 LINE.
MOVE HDR-4 TO P-REC.
PERFORM INSP THRU EXT5.
WRITE P-REC AFTER 2 LINE.
EXTP.
EXIT.
* الفقرة التالية هي لطباعة المجاميع
FINAL-RTN.
MOVE SPACES TO P-REC.
WRITE P-REC AFTER ADVANCING PAGE.
MOVE SPACES TO P-REC.
MOVE FIN-1 TO P-REC.
PERFORM INSP THRU EXT5.
WRITE P-REC AFTER 30.
MOVE CNTR1 TO PCNTR1.
MOVE FIN-2 TO P-REC.
PERFORM INSP THRU EXT5.
WRITE P-REC AFTER 2.
MOVE CNTR2 TO PCNTR2.
MOVE FIN-3 TO P-REC.
PERFORM INSP THRU EXT5.
WRITE P-REC AFTER 2.
MOVE CNTR3 TO PCNTR3.
MOVE FIN-4 TO P-REC.
PERFORM INSP THRU EXT5.
WRITE P-REC AFTER 2.
EXTF.
EXIT.
* يتم تحويل الأرقام من لاتيني إلى عربي
INSP.
INSPECT P-REC REPLACING ALL ' ' BY ' '
INSPECT P-REC REPLACING ALL '0' BY '٠'
INSPECT P-REC REPLACING ALL '1' BY '١'
INSPECT P-REC REPLACING ALL '2' BY '٢'
INSPECT P-REC REPLACING ALL '3' BY '٣'
INSPECT P-REC REPLACING ALL '4' BY '٤'
INSPECT P-REC REPLACING ALL '5' BY '٥'
INSPECT P-REC REPLACING ALL '6' BY '٦'
INSPECT P-REC REPLACING ALL '7' BY '٧'
INSPECT P-REC REPLACING ALL '8' BY '٨'
INSPECT P-REC REPLACING ALL '9' BY '٩'.
EXT5.
EXIT.

```

١٣ - ٥ تمارين عامة :

تمرين رقم (١) :

اكتب برنامجاً بلغة كويول يطبع في يسار سطر الطباعة الأرقام من ١ - ١٠ بعدد رقم في كل سطر، وضمن خمسة مجموعات، واترك سطر فراغ بين كل مجموعتين.

— استخدام تعليمة **PERFORM** —

شكل الاخراج :

1
2
3
4 مجموعة ١
5
6
7
8
9
10

1
2
3
4 مجموعة ٢

| | |
|-----|-----|
| ٥٥ | ٥٥ |
| ٥٦ | ٥٦ |
| ٥٧ | ٥٧ |
| ٥٨ | ٥٨ |
| ٥٩ | ٥٩ |
| ٦٠ | ٦٠ |
| ٦١ | ٦١ |
| ٦٢ | ٦٢ |
| ٦٣ | ٦٣ |
| ٦٤ | ٦٤ |
| ٦٥ | ٦٥ |
| ٦٦ | ٦٦ |
| ٦٧ | ٦٧ |
| ٦٨ | ٦٨ |
| ٦٩ | ٦٩ |
| ٧٠ | ٧٠ |
| ٧١ | ٧١ |
| ٧٢ | ٧٢ |
| ٧٣ | ٧٣ |
| ٧٤ | ٧٤ |
| ٧٥ | ٧٥ |
| ٧٦ | ٧٦ |
| ٧٧ | ٧٧ |
| ٧٨ | ٧٨ |
| ٧٩ | ٧٩ |
| ٨٠ | ٨٠ |
| ٨١ | ٨١ |
| ٨٢ | ٨٢ |
| ٨٣ | ٨٣ |
| ٨٤ | ٨٤ |
| ٨٥ | ٨٥ |
| ٨٦ | ٨٦ |
| ٨٧ | ٨٧ |
| ٨٨ | ٨٨ |
| ٨٩ | ٨٩ |
| ٩٠ | ٩٠ |
| ٩١ | ٩١ |
| ٩٢ | ٩٢ |
| ٩٣ | ٩٣ |
| ٩٤ | ٩٤ |
| ٩٥ | ٩٥ |
| ٩٦ | ٩٦ |
| ٩٧ | ٩٧ |
| ٩٨ | ٩٨ |
| ٩٩ | ٩٩ |
| ١٠٠ | ١٠٠ |

تمرين رقم (٢) :

اكتب برنامجاً بلغة كوبرول بحيث يقرأ سجل زبون — عميل — CUSTOMER في بنك تجاري ويحسب له الرصيد الجديد؛ علماً بأن أسماء الزبائن موجودة على ملف أساسي متسلسل SEQUENTIAL على قرص ممغنط ، واسم الملف DEFFILE — انظر توصيف الملف في الصفحة التالية :

مع مراعاة ما يلي :

- ١ — بعد قراءة كل سجل تحقق من حقل رمز السجل D-CODE فإذا كان هذا الحقل يحتوي على الحرف 'D' فهذا يعني أن السجل ملغى فتجاهله ولا تعالجه .
- ٢ — أما إذا كان الحقل D-CODE يحتوي على الرمز 'A' فهذا يعني أن السجل

تحت المعالجة ACTIVE فأجر عليه حساب الرصيد الجديد وذلك حسب الخطوات التالية :

NEW-BAL الرصيد الجديد = رصيد آخر المدة D-ENDBAL + المبلغ المودع لتاريخه
D-DPST — المبلغ المسحوب لتاريخه D-DRFT

$$\text{NEW-BAL} = \text{D-ENDBAL} + \text{D-DPST} - \text{D-DRFT}$$

بعد اجراء عملية الحساب السابقة اطبع محتوى السجل مع الرصيد الجديد على الطابعة وكما هو مبين في النموذج المرفق الخاص بالطباعة .

٣ — كرر العمل السابق من أجل كل سجل .

٤ — في نهاية العمل اطبع :

مجموع المودعين — عدا المحنوفين

مجموع الأرصدة القديمة

مجموع المبالغ المودعة

مجموع المبالغ المسحوبة

مجموع الأرصدة الجديدة

مع مراعاة ما يلي :

١ — اطبع التاريخ ورقم الصفحة برأس كل صفحة .

٢ — اترك فراغاً بين كل سطري كتابة .

٣ — لا تطبع في الصفحة الواحدة أكثر من ٢٢ سطر .

٤ — اطبع المجاميع النهائية على صفحة مستقلة .

٥ — استعمل الفاصلة (,) بدلاً من (.) للدلالة على الفاصلة العشرية .

أظهر الاشارة الجبرية للرصيد الجديد على يسار حقل طباعة هذا الحقل .

توصيف ملف الادخال : DEFILE

| FIELD NAME | PICTURE |
|-------------------------------|----------------------------|
| رمز السجل D-CODE | X |
| رقم الحساب D-ACNO | 9 (5) |
| اسم الزبون D-ACNAME | X (34) |
| رصيد آخر المدة D-ENDBAL | 9 (7) V99 أنظر الصفحة رقم |
| المبلغ المودع لتاريخه D-DPST | 9 (7) V99 ٦٠٩ لتوصيف شكل |
| المبلغ المسحوب لتاريخه D-DRFT | 9 (7) V99 المخرجات والصفحة |
| فراغ FILLER | X (17) رقم ٦٠٩ للنتائج |

تسرين رقم (٣) :

لديك ملف اسمه VALFILE وهو ملف متسلسل SEQUENTIAL موجود على قرص ممغنط ويحتوي على أسماء وأرقام ورمز المنطقة بالاضافة إلى معلومات أخرى .
 المطلوب كتابة برنامج يقرأ هذا الملف و يعالجه ، وفي نهاية العمل اطبع جدولاً تكرارياً يحتوي على رمز المنطقة واسم المنطقة وعدد المشتركين في كل منطقة والمجموع الكلي للمشاركين . حسب الشكل التالي :

| الرمز | اسم المنطقة | عدد المشتركين |
|-------|-------------|---------------|
| Z9 | XXXXXXXX | ZZ9 |
| Z9 | XXXXXXXX | ZZ9 |
| . | | . |
| . | | . |
| . | | . |
| . | | . |
| . | | . |
| ** | العند الكلي | ZZ99 |

علماً بأن توصيف السجل كالتالي :

| FIELD NAME | PICTURE |
|----------------------|---------|
| VAL-NO | 9 (5) |
| VAL-NAME | X (30) |
| VAL-ADRES | X (24) |
| VAL-DATE | 9 (8) |
| VAL-CODE | 99 |
| رمز المنطقة VAL-AREA | 99 |
| BLNK | 9(8) |

وأن رموز المناطق كالتالي :

| الرمز | اسم المنطقة |
|-------|-------------|
| ٠١ | الرياض |
| ٠٢ | جدة |
| ٠٣ | الدمام |
| ٠٤ | القصيم |
| ٠٥ | حائل |
| ٠٦ | المدينة |
| ٠٧ | أبها |
| ٠٨ | بلجرشي |
| ٠٩ | عسير |
| ١٠ | شقرة |

تعرين رقم (٤) :

تريد شركة صناعية احتساب الأجر الإضافي لعمالها، علماً بأن جميع سجلات
RECORDS العمال موجودة على ملف مفهرس INDEXED FILE اسمه PAYMST
واسم السجل فيه RECMST وموجود على قرص ممغنط DISC وشكل السجل
فيه كالتالي :

| FIELD NAME | PICTURE |
|--------------------------------|------------|
| رقم العامل (KEY).... EMP-NO | X (5) |
| اسم العامل EMP-NAME | X (34) |
| أجره الشهري EMP-SALARY | 9 (5) V 99 |
| EMP-HORS1 | 999 |
| EMP-HORS2 | 999 |
| EMP-HORS3 | 999 |
| مجموع الأجر الإضافي EMP-OVTIME | 9 (5) V99 |
| EMP-FLER | X (22) |

حيث أن :

- EMP-SALARY : يمثل الأجر الشهري للعامل بمعدل ٨ ساعات عمل يومياً في ٣٠ يوم للشهر.
- EMP-HORS1 : تمثل عدد الساعات التي تحسب أجرتها وكأنها ساعة عمل واحدة لكل ساعة.
- EMP-HORS2 : تمثل عدد الساعات التي تحسب أجرتها وكأنها ساعة ونصف عمل عادي لكل ساعة.

- EMP-HORS3 : تمثل عدد الساعات التي تحسب أجرتها وكأنها ساعتي عمل عادي لكل ساعة .
- EMP-OVTIME : مبلغ الأجر الاضافي المطلوب حسابه .

والمطلوب ما يلي :

كتابة برنامج يحسب الأجر الاضافي EMP-OVTIME لكل عامل و يكتبه على نفس السجل بشرط ألا يتجاوز الأجر الاضافي مقدار نصف الأجر الشهري :

و يتم العمل وفق الخطوات التالية :

- ١ — يقرأ الملف تسلسلياً .
- ٢ — يجري حساب أجرة الساعة الواحدة .
- ٣ — تضرب أجرة الساعة الواحدة بعدد ساعات العمل الاضافي على التوالي ولثلاثة الأنواع السابقة :

EMP-HORS3 , EMP-HORS2 , EMP-HORS1

- ٤ — يقارن الناتج مع مقدار نصف الراتب :
- فإذا كان الناتج أصغر أو مساوٍ لنصف الراتب ينقل الناتج إلى حقل

EMP-OVTIME

- أما إذا كان أكبر من نصف الراتب فينقل مقدار نصف الراتب إلى حقل EMP-OVTIME واطبع على الطباعة رقم الموظف واسمه وراتبه الأساسي وعبارة «تجاوز الحد الأعلى» .

٥ — اعد كتابة السجل على القرص .

٦ — كرر العمل من جديد .

٧- في نهاية العمل أطلع :

مجموع الأجور الشهرية للعمال : 9999999.99

مجموع بدلات خارج الدوام : 9999999.99

عدد العمال الكلي : 9999

٨- انهي العمل . أنظر الصفحة رقم ٦١٠ لتوصيف شكل المخرجات.

تمرين رقم (٥) :

لديك ملفان متسلسلان ومفروزان تصاعدياً على نفس الحقل وهو رقم المشترك .

الملف الأول واسمه SUBFILE ملف المشتركين وشكل سجله :

| | FIELD NAME | PICTURE |
|-----------------|------------|---------|
| رقم المشترك | SUB-NO | 9 (5) |
| اسمه | SUB-NAME | X (24) |
| عنوانه | SUB-ADRESS | X (24) |
| تاريخ الاشتراك | SUB-DATE | 9 (8) |
| رمز الاشتراك | SUB-CODE | 99 |
| رمز المنطقة | SUB-AREA | 99 |
| رقم العدد | SUB-CNTR | 9 (7) |
| مقدار آخر قراءة | SUB-LREAD | 9 (7) |
| فراغ | SUB-BLNK | X (5) |

والملف الثاني واسمه CONSMFPL وهو ملف الاستهلاك وشكل سجله :

| | FIELD NAME | PICTURE |
|-----------------------|------------|---------|
| رقم المشترك | CONS-NO | 9 (5) |
| مقدار القراءة الحالية | CONS-READ | 9 (7) |

| | | |
|---------------|------------|--------|
| تاريخ القراءة | CONS-DATE | 9 (8) |
| ملاحظات | CONS-COMNT | X (34) |
| فراغ | FILLER | X (34) |

المطلوب قراءة الملفين السابقين والتأكد أنه لكل سجل في ملف الاستهلاك CONSMPL هناك سجل مقابل في ملف المشتركين الأساسي SUBFILE وفي حال وجود سجل أساسي انقل سجل الاستهلاك إلى ملف آخر اسمه NCONSMPL وشكل سجله مطابق لسجل ملف الاستهلاك، أما إذا كان السجل في ملف الاستهلاك لا يحتوي مقابل له في سجل المشتركين اطبع محتوى هذا السجل على الطباعة - حسب الشكل المرفق - ولا تنقله على ملف الاستهلاك الجديد NCONSMPL .
انهي العمل عند الانتهاء من ملف الاستهلاك . أنظر شكل المخرجات . صفحة ٦١١

تسرين رقم (٦) :

لدينا ملف على شريط ممغنط اسمه STDFILE يحتوي على أرقام وبيانات طلاب إحدى الجامعات في المملكة ومفروزة تصاعدياً حسب رمز الكلية ورقم السنة ضمن كل كلية .

المطلوب : كتابة برنامج يطبع سجل كل طالب ضمن كل سنة للكلية الواحدة

وايجاد :

١ - مجموع طلاب كل سنة ضمن كل كلية MINOR TOTAL

٢ - مجموع طلاب كل كلية ضمن الجامعة MAJOR TOTAL

٣ - مجموع طلاب الجامعة OVERALL TOTAL

• لا تكتب في الصفحة أكثر من ٣٠ سطراً .

•• عند طباعة المجاميع اترك سطرين فارغين قبل وبعد سطر المجاميع .

••• انظر الصفحة المرفقة من أجل توصيف الطباعة .

شكل سجل الملف STDFILE :

| | FIELD NAME | PICTURE |
|--------------------|------------|---------|
| رقم الطالب | STD-NO | 9 (5) |
| اسم الطالب | STD-NAME | X (34) |
| رمز الكلية | STD-CLG | 99 |
| رقم السنة الدراسية | STD-YEAR | 99 |
| معلومات | STD-DATA | X (11) |

مفتاح الحل :

- ١ - عين عدادات COUNTERS من أجل المجاميع.
- ٢ - عين مساحات عمل WORK AREAS من أجل مقارنة رمز الكلية أو السنة مع رمز الكلية أو السنة للسجل السابق.
- أنظر الصفحة رقم ٦١٢ لتوصيف شكل المخرجات.

الفصل الرابع عشر

المراجع

١٤-١. المراجع العربية :

- ١ - المعجم العربي الموحد لمصطلحات الحاسبات الألكترونية، المنظمة العربية للعلوم الادارية - جامعة الدول العربية، عمان ١٩٨١م.
- ٢ - الحاسبات الألكترونية وسيلة لتطوير المجتمع، د. أمين الصالح، وزارة الثقافة، دمشق ١٩٨١م.
- ٣ - المبادئ الأساسية في الحاسبات الألكترونية، السيد محمد السيد، دار المعارف، القاهرة ١٩٧٣م.
- ٤ - أربعة مداخل للحاسبات الألكترونية، أحمد المطاوع، عمان ١٩٧٨م.
- ٥ - قاموس شرح المصطلحات العلمية للأجهزة المحاسبة الآلية، د. حسام البيلاوي، منشأة المعارف، الاسكندرية ١٩٧٦م.

١٤-٢. المراجع الأجنبية :

1. Davis, W.S. and Fisher, R.H. "An Introduction to Structured Logic and Modular Program Design", Addison. Wesley, 1979.
2. Grauer, Robert T. "A Cobol Book of Practice and Reference", Prentice - Hall, N.J., 1981.
3. Grauer, Robert T. "Cobol a Vehicle for Information Systems", Prentice - Hall, N.J., 1981.

4. Grauer, Robert T. and Crawford, Marshal A. **"Structured Cobol, a Pragmatic Approach"**, Prentice - Hall N.J., 1981.
5. Harrison, W.S. **"A Programmer's Guide to Cobol"**, V.N.R., 1980.
6. McClure Carma **"Reducing Cobol Complexity Through Structured Programming"**, V.N.R., 1978.
7. N.C.C. **"Cobol Reference Summary"** 1980 Manchester, 1980.
8. N.C.C. **"Cobol : Student Notes and Work Book"**, Manchester, 1980.
9. N.C.C. **"Introducing Systems' Analysis and Design"**, Vol. 1 and 2, 1980.
10. N.C.R. Corp. **"NCR IMOS Cobol : Student Text"**, 4th Edition, Ohio, 1981.
11. IBM Corp. **"IBM OS VS Cobol compiler and Library, Programmer's Guide"** SC 28-6483-2 Oct. 1981.
12. IBM Corp. **"American National Standard Cobol, Coding, Illustration's"**, Feb., 1978.
13. IBM Corp. **"American National Standard Cobol Coding : Coding Techniques and Disk Applications Text"**, May, 1978.
14. IBM Corp. **"Time Sharing Option, 3270 Display Support and Structured Programming Facilities"**, V. 2.2, Program Ref. Manual, SH 20-1975-2 July, 1979.
15. Parkin, Arnold **"Cobol for Students"**, Arnold, 1975.
16. Philippakis A.S. and Kazmier L.J. **"Advanced Cobol"**, McGraw Hill, 1982.
17. Popkin, Gary S. **"Introductory Structured Cobol Programming"**, V.N.R. N.Y., 1981.
18. Philippakis, A.S. and Kazmier, L.J. **"Structured Cobol"**, McGraw-Hill 1981.
19. Shelly, G.B. and Cashman, T.J. **"Dos Job Control for Cobol Programmers"** Anheim, 1977.
20. Shelly, G.B. and Cashman, T.J. **"Introduction to Flowcharting and Computer**

Programming Logic", Anaheim, 1978, 9th ed.

21. Silver and Silver **"Computer Algorithms and Flowcharting"** McGraw-Hill, 1975.

22. Sordillo Donald A. **"The Programmer's ANSI Cobol Reference Manual"**, Prentice - Hall, N.J., 1978.

23. Stern/Stern **"Structured Cobol Programming"**, John Wiley, 1980, 3rd ed.

24. Stuart, F. **"Introduction to Standard Cobol Programming"**, H.B.J. 1974.

25. Texas Instruments, **"Model 990 Computer Cobol Reference Manual"**, 1980.

26. IBM Corp., **'OS/VS2 MVS JCL'**, GC28/0692/4, May 1979.

الفصل الخامس عشر ملحق

١٥-١ جدول رموز الاخراج *

| First Symbol | Nonflaring Insertion Symbols | | | | | | | | | | Flaring Insertion Symbols | | | | | | | | | | Other Symbols | | | | |
|------------------------------|------------------------------|---|---|---|-------------------------|-------------------------|------------|---|------------|------------|---------------------------|-------------------------|----------------|----------------|---|---|---|---|----------------|----------------|---------------|---|--|--|--|
| Second Symbol | B | D | / | . | (+) ¹ [-] | (+) ¹ [-] | CR [DB] | a | [Z] [a] | [Z] [a] | (+) ¹ [-] | (+) ¹ [-] | a ² | a ² | q | A | S | V | P ¹ | P ² | | | | | |
| Nonflaring Insertion Symbols | B | x | x | x | x | x | | x | x | x | x | x | x | x | x | x | | | | | x | | | | |
| | D | x | x | x | x | x | | x | x | x | x | x | x | x | x | x | | | | | x | | | | |
| | / | x | x | x | x | x | | x | x | x | x | x | x | x | x | | | | | x | | | | | |
| | . | x | x | x | x | x | | x | x | x | x | x | x | x | | | | | | | | | | | |
| | (+) ¹ [-] | | | | | | | x | x | | x | x | | | | | | | | | | | | | |
| | (+) ¹ [-] | x | x | x | x | x | | x | x | x | | | x | x | | | | | | | x | x | | | |
| | CR [DB] | x | x | x | x | x | | x | x | x | | | x | x | | | | | | | x | x | | | |
| Flaring Insertion Symbols | a | | | | | x | | | | | | | | | | | | | | | | | | | |
| | [Z] [a] | x | x | x | x | x | | x | x | | | | | | | | | | | | | | | | |
| | [Z] [a] | x | x | x | x | x | | x | x | x | | | | | | | | | | | x | | | | |
| | (+) ¹ [-] | x | x | x | x | | | x | | | x | | | | | | | | | | | | | | |
| | (+) ¹ [-] | x | x | x | x | x | | x | | | x | x | | | | | | | | | x | x | | | |
| | a ² | x | x | x | x | x | | | | | | | x | | | | | | | | | | | | |
| Other Symbols | a ² | x | x | x | x | x | | | | | | | x | x | | | | | | | x | | | | |
| | V | x | x | x | x | x | | x | x | x | x | x | x | x | x | x | x | x | x | | | | | | |
| | A | x | x | x | | | | | | | | | | | | | | | | | | | | | |
| | X | x | x | x | | | | | | | | | | | | | | | | | | | | | |
| | S | | | | | | | | | | | | | | | | | | | | | | | | |
| | V | x | x | x | x | x | | x | x | x | x | x | x | x | x | x | x | x | x | | | | | | |
| | P ¹ | x | x | x | x | x | | x | x | x | x | x | x | x | x | x | x | x | x | | | | | | |
| | P ² | | | | | | | x | | | | | | | | | | | | | x | x | | | |

¹ When character is to left of decimal point
² When character is to right of decimal point

١٥ - ٢ الكلمات المحجوزة في لغة كوبول *

* COBOL RESERVED WORDS

| | | |
|---------------------|------------------|---------------|
| ACCEPT | BOOLEAN | CONNECT |
| ACCESS | BOTTOM | *CONSOLE |
| *ACTUAL | BY | CONTAINS |
| ADD | | CONVERSION |
| ADDRESS | CALL | CONVERTING |
| ADVANCING | *CANCEL | CONTROL |
| AFTER | CBL | CONTROLS |
| ALPHABET | CD | COPY |
| ALL | CF | *CORE-INDEX |
| ALPHABETIC | CH | CORR |
| ALPHANUMERIC | *CHANGED | CORRESPONDING |
| ALPHANUMERIC-EDITED | CHARACTER | COUNT |
| ALSO | CHARACTERS | *CSP |
| ALTER | CLOCK-UNITS | CURRENCY |
| ALTERNATE | CLOSE | *CURRENT-DATE |
| AND | COBOL | *CYL-INDEX |
| ANY | CODE | *CYL-OVERFLOW |
| *APPLY | CODE-SET | *C01 |
| ARE | COLUMN | *C02 |
| AREA | *COM-REG | *C03 |
| AREAS | COMMA | *C04 |
| ASCENDING | COMP | *C05 |
| ASSIGN | *COMP-1 | *C06 |
| AT | *COMP-2 | *C07 |
| AUTHOR | *COMP-3 | *C08 |
| | *COMP-4 | *C09 |
| *BASIS | COMPUTATIONAL | *C10 |
| BEFORE | *COMPUTATIONAL-1 | *C11 |
| *BEGINNING | *COMPUTATIONAL-2 | *C12 |
| BIT | *COMPUTATIONAL-3 | |
| BITS | *COMPUTATIONAL-4 | DATA |

• Structured Cobol, Grimmer and Crawford. PP 366-369

| | | |
|----------------|------------------|----------------|
| BLANK | COMPUTE | *DATE |
| BLOCK | CONFIGURATION | DATE-COMPILED |
| DATE-WRITTEN | ENTER | INPUT-OUTPUT |
| *DAY | *ENTRY | *INSERT |
| DAY-OF-WEEK | ENVIRONMENT | INSPECT |
| DB-CONFLICT | EOP | INSTALLATION |
| DB-EXCEPTION | EQUAL | INTO |
| DB-KEY | EQUALS | INVALID |
| DB-PRIVATE-KEY | ERASE | IS |
| DB-REALM-NAME | ERROR | |
| DB-RECORD-NAME | ESI | JUST |
| DB-SET-NAME | EVERY | JUSTIFIED |
| DB-STATUS | EXCEEDS | |
| DE | EXCEPTION | KEEP |
| *DEBUG | EXCLUSIVE | KEY |
| DEBUG-CONTENTS | *EXHIBIT | |
| DEBUG-ITEM | EXIT | LABEL |
| DEBUG-LENGTH | EXTEND | *LABEL-RETURN |
| DEBUG-LINE | *EXTENDED-SEARCH | LAST |
| DEBUG-NAME | | LEADING |
| DEBUG-NUMERIC- | FD | *LEAVE |
| CONTENTS | FILE | LEFT |
| DEBUG-SIZE | FILE-CONTROL | LENGTH |
| DEBUG-START | FILES | LESS |
| DEBUG-SUB | FILLER | LIMIT |
| DEBUG-SUB-ITEM | FINAL | LIMITS |
| DEBUG-SUB-N | FIND | LINAGE |
| DEBUG-SUB-NUM | FINISH | LINAGE-COUNTER |
| DEBUG-SUB-1 | FIRST | LINE |
| DEBUG-SUB-2 | FOOTING | LINE-COUNTER |
| DEBUG-SUB-3 | FOR | LINES |
| DEBUGGING | FREE | LINKAGE |
| DECIMAL-POINT | FROM | LOCALLY |
| DECLARATIVES | | LOCK |
| DELETE | GENERATE | LOW-VALUE |
| DELIMITED | GET | LOW-VALUES |
| DELIMITER | GIVING | |
| DEPENDING | GO | *MASTER-INDEX |
| DEPTH | *GOBACK | MEMBER |
| DESCENDING | GREATER | MEMBERS |
| DESTINATION | GROUP | MEMBERSHIP |
| DETAIL | | MEMORY |
| DISABLE | HEADING | MERGE |
| *DISP | HIGH-VALUE | MESSAGE |
| DISPLAY | HIGH-VALUES | MODE |
| *DISPLAY-ST | | MODIFY |

| | | |
|-----------------|--------------------|-----------------|
| DIVIDE | I-O | MODULES |
| DIVISION | I-O-CONTROL | *MORE-LABELS |
| DOWN | *ID | MOVE |
| DUPLICATE | IDENTIFICATION | MULTIPLE |
| DUPLICATES | IF | MULTIPLY |
| DYNAMIC | IN | |
| | INCLUDING | *NAMED |
| EGI | INDEX | NATIVE |
| *EJECT | INDEX-N | NEGATIVE |
| ELSE | INDEXED | NEXT |
| EMI | INDICATE | NO |
| ENABLE | INITIAL | *NOMINAL |
| END | INITIALIZE | NON-NULL |
| END-OF-PAGE | INITIATE | NOT |
| *ENDING | INPUT | *NSTD-REELS |
| NULL | REALM | *SKIP1 |
| NUMBER | REALMS | *SKIP2 |
| NUMERIC | REALM-NAME | *SKIP3 |
| NUMERIC-EDITED | *RECEIVE | SORT |
| | RECORD | *SORT-CORE-SIZE |
| OBJECT-COMPUTER | RECORD-NAME | *SORT-FILE-SIZE |
| OCCURS | *RECORD-OVERFLOW | *SORT-MERGE |
| OF | RECORDS | *SORT-MESSAGE |
| OFF | REDEFINES | *SORT-MODE-SIZE |
| OMITTED | REEL | SORT-OPTION |
| ON | REFERENCE-MODIFIER | *SORT-RETURN |
| ONLY | REFERENCES | SOURCE |
| OPEN | RELATIVE | SOURCE-COMPUTER |
| OPTIONAL | RELEASE | SPACE |
| OR | *RELOAD | SPACES |
| ORDER | REMAINDER | SPECIAL-NAMES |
| ORGANIZATION | REMONITOR | STANDARD |
| OTHER | REMOVAL | STANDARD-1 |
| *OTHERWISE | RENAMES | STANDARD-2 |
| OUTPUT | *REORG-CRITERIA | START |
| OVERFLOW | REPLACING | STATUS |
| OWNER | REPORT | STOP |
| | REPORTING | STORE |
| PADDING | REPORTS | STRING |
| PAGE | *REREAD | SUB-QUEUE-1 |
| PAGE-COUNTER | RERUN | SUB-QUEUE-2 |
| *PASSWORD | RESERVE | SUB-QUEUE-3 |
| PERFORM | RESET | SUB-SCHEMA |
| PERMANENT | RETAINING | SUBTRACT |
| PF | RETRIEVAL | SUM |
| PH | RETURN | SUPPRESS |

| | | |
|---------------|---------------|-----------------|
| PIC | •RETURN-CODE | SUSPEND |
| PICTURE | REVERSED | SYMBOLIC |
| PLUS | REWIND | SYNC |
| POINTER | REWRITE | SYNCHRONIZED |
| POSITION | RF | *SYSIN |
| •POSITIONING | RH | *SYSIPT |
| POSITIVE | RIGHT | *SYSLSLST |
| •PRINT-SWITCH | ROUNDED | *SYSOUT |
| PRINTING | RUN | *SYSPCH |
| PRIOR | | *SYSPUNCH |
| PRIVACY | SAME | *S01 |
| PROCEDURE | SD | *S02 |
| PROCEDURES | SEARCH | *S03 |
| PROCEED | SECTION | *S04 |
| PROCESS | SECURITY | *S05 |
| •PROGRAM | SEGMENT | |
| PROGRAM-ID | SEGMENT-LIMIT | TABLE |
| PROTECTED | SELECT | TALLY |
| PURGE | SELECTIVE | TALLYING |
| | SEND | TAPE |
| QUEUE | SENTENCE | TENANT |
| QUOTE | SEPARATE | TERMINAL |
| QUOTES | SEQUENCE | TERMINATE |
| | SEQUENTIAL | TEXT |
| | *SERVICE | THAN |
| RANDOM | SET | *THEN |
| RD | SIGN | THROUGH |
| READ | SIZE | THRU |
| •READY | UNSTRING | VALUE |
| TIME | UNTIL | VALUES |
| •TIME-OF-DAY | UP | VARYING |
| TIMES | UPDATE | |
| TO | UPON | WHEN |
| TOP | *UPSI-0 | •WHEN-COMPILED |
| •TOTALLED | *UPSI-1 | WITH |
| •TOTALING | *UPSI-2 | WITHIN |
| •TRACE | *UPSI-3 | WORDS |
| •TRACK | *UPSI-4 | WORKING-STORAGE |
| •TRACK-AREA | *UPSI-5 | WRITE |
| •TRACK-LIMIT | *UPSI-6 | •WRITE-ONLY |
| TRACKS | *UPSI-7 | •WRITE-VERIFY |
| TRAILING | USAGE | |
| TRANSFORM | USAGE-MODE | ZERO |
| TYPE | USE | ZEROES |
| UNEQUAL | USING | ZEROS |
| UNIT | | |

• Note: This set of reserved words is based on the ANS 74 standard and CODASYL Journal of

Development January 1976. IBM extensions are identified with an asterisk.

٣-١٥ الصيغ العامة في لغة كوبرول *

GENERAL FORMATS

General Format For Identification Division

IDENTIFICATION DIVISION
PROGRAM-ID = program-name
AUTHOR = [comment-entry] ...
INSTALLATION = [comment-entry] ...
DATE-WRITTEN = [comment-entry] ...
DATE-COMPILED = [comment-entry] ...
SECURITY = [comment-entry] ...

General Format For Environment Division

ENVIRONMENT DIVISION
CONFIGURATION SECTION
SOURCE-COMPUTER = computer-name [WITH DEBUGGING MODE]
OBJECT-COMPUTER = computer-name

[MEMORY SIZE integer { WORDS
CHARACTERS
MODULES }]

[PROGRAM COLLATING SEQUENCE IS alphabet-name]

[SEGMENT-LIMIT IS segment-number]

*[SPECIAL-NAMES = [implementor-name

{ IS mnemonic-name [ON STATUS IS condition-name-1 [OFF STATUS IS condition-name-2]]
IS mnemonic-name [OFF STATUS IS condition-name-2 [ON STATUS IS condition-name-1]] ...
ON STATUS IS condition-name-1 [OFF STATUS IS condition-name-2]
OFF STATUS IS condition-name-2 [ON STATUS IS condition-name-1]

[alphabet-name IS { STANDARD-1
NATIVE
implementor-name
{ literal-1 { THROUGH
THRU } literal-2 }
ALSO literal-3 [ALSO literal-4] ... }] ...

* المصدر :

Donald A. Sordillo

The Programmer's ANSI Cobol Reference Manual, Prentice - Hall, Inc, New Jersey, 1978,

pp. 365 - 378.

```

*INPUT-OUTPUT SECTION*
  FILE-CONTROL.
    {file-control-entry}...
*I-O-CONTROL*

```

```

[RERUN ON [file-name-1
  implementor-name-1]] EVERY {
  {END OF [REEL
    UNIT]} OF file-name-2 }n ...
  integer-1 RECORDS
  integer-2 CLOCK-UNITS
  condition-name
}

[SAME [RECORD
  SORT
  SORT-MERGE]] AREA FOR (file-name-3) ...

```

N.B. The superscripts above the brackets are to indicate matching sets; they are not part of the COBOL language.

Format 1

Format 2

```
[ACCESS MODE IS { SEQUENTIAL (RELATIVE KEY IS data-name-1)
                  RANDOM
                  DYNAMIC } RELATIVE KEY IS data-name-1)]
(FILE STATUS IS data-name-2)*
```

Format 3

```

SELECT file-name
  ASSIGN TO {implementor-name-1} ...
  {RESERVE integer-1 {AREA
    {AREAS}}}
  ORGANIZATION IS INDEXED
  {ACCESS MODE IS {SEQUENTIAL}
    {RANDOM
    DYNAMIC}}
  RECORD KEY IS data-name-1
  [ALTERNATE RECORD KEY IS data-name-2 {WITH DUPLICATES}].
  FILE STATUS IS data-name-5]

```

Format 4

SELECT file-name **ASSIGN TO** (implementor-name-1) ...

General Format For Data Division

```

DATA DIVISION.
FILE SECTION.
FD file-name
  [BLOCK CONTAINS {integer-1 TO} integer-2 {RECORDS
  CHARACTERS}]
  [RECORD CONTAINS {integer-3 TO} integer-4 CHARACTERS]
  LABEL {RECORD IS {STANDARD}
  RECORDS ARE {OMITTED}
  VALUE OF {implementor-name-1 IS {data-name-1}} ...}
  [DATA {RECORD IS
  RECORDS ARE} {data-name-3} ...]
  [LINAGE IS {data-name-5}
  integer-5] LINES [WITH FOOTING AT {data-name-6}
  integer-6]
  [LINES AT TOP {data-name-7}
  integer-7][LINES AT BOTTOM {data-name-8}
  integer-8]
  [CODE SET IS alphabet-name]
  [REPORT IS
  REPORTS ARE] {report-name-1} ...].
[record-description-entry] ... ] ...
SD file-name
  [RECORD CONTAINS {integer-1 TO} integer-2 CHARACTERS]
  [DATA {RECORD IS
  RECORDS ARE} {data-name-1}] =
  [record-description-entry] ... ] ...

```


[WORKING-STORAGE SECTION*

[77-level-description-entry]
[record-description-entry] ...]

[LINKAGE SECTION*

[77-level-description-entry]
[record-description-entry] ...]

[COMMUNICATION SECTION*

[communication-description-entry
[record-description-entry] ...] ...]

REPORT SECTION

TRD report-name

[CODE literal-1]

[[CONTROL IS] { [data-name-1] ... }]
[[CONTROLS ARE] [FINAL [data-name-1] ...]]

[[PAGE [LIMIT IS integer-1] [LINE] [LIMITS ARE] integer-1] [HEADING integer-2]

[[FIRST DETAIL integer-3] [LAST DETAIL integer-4]

[[FOOTING integer-5]]*

[(report-group-description-entry) ...]* ...]*

N.B. The superscripts above the brackets are to indicate matching sets; they are not part of the COBOL language.

General Format For Data Description Entry

Format 1

level-number { data-name-1 }
[FILLER]

[REDEFINES data-name-2]

[{ PICTURE] IS character-string]
[PIC]

[[USAGE IS] { [COMPUTATIONAL]
[COMP]
[DISPLAY]
[INDEX] }]

[(SIGN IS) { LEADING] [SEPARATE CHARACTER]
[TRAILING]]

[OCCURS { integer-1 TO integer-2 TIMES DEPENDING ON data-name-3 }
integer-2 TIMES]

[[ASCENDING]
[DESCENDING] KEY IS { data-name-4 } ...]

[INDEXED BY { index-name-1 } ...]

[{ SYNCHRONIZED] [LEFT]
[SYNC] [RIGHT]]

[{ JUSTIFIED] RIGHT]
[JUST]

[BLANK WHEN ZERO]

[VALUE IS literal]*

Format 2

66 data-name-1 RENAMES data-name-2 { THROUGH } data-name-3 }.

Format 3

88 condition-name { VALUE IS } { VALUES ARE } { literal-1 { THROUGH } literal-2 } ... *

General Format For Communication Description Entry

Format 1

CD cd-name

FOR [INITIAL] INPUT

| | | | | |
|--|---|---|---|---|
| | [| { | [SYMBOLIC <u>QUEUE</u> IS data-name-1] | } |
| | | | [SYMBOLIC SUB- <u>QUEUE-1</u> IS data-name-2] | |
| | | | [SYMBOLIC SUB- <u>QUEUE-2</u> IS data-name-3] | |
| | | | [SYMBOLIC SUB- <u>QUEUE-3</u> IS data-name-4] | |
| | | | [<u>MESSAGE DATE</u> IS data-name-5] | |
| | | | [<u>MESSAGE TIME</u> IS data-name-6] | |
| | | | [SYMBOLIC <u>SOURCE</u> IS data-name-7] | |
| | | | [<u>TEXT LENGTH</u> IS data-name-8] | |
| | | | [<u>END KEY</u> IS data-name-9] | |
| | | | [<u>STATUS KEY</u> IS data-name-10] | |
| | | | [<u>MESSAGE COUNT</u> IS data-name-11]]* | |
| | | | [data-name-1, data-name-2, ..., data-name-11] |] |

N.B. The superscripts above the brackets are to indicate matching sets; they are not part of the COBOL language.

Format 2

CD cd-name FOR OUTPUT

[DESTINATION COUNT IS data-name-1]

[TEXT LENGTH IS data-name-2]

[STATUS KEY IS data-name-3]

[DESTINATION TABLE OCCURS integer-2 TIMES [INDEXED BY {index-name-1}...]]

[ERROR KEY IS data-name-4]

[SYMBOLIC DESTINATION IS data-name-5]

General Format For Report Group Description Entry

Format 1

01 {data-name-1}

[LINE NUMBER IS {integer-1 [ON NEXT PAGE]}]

[NEXT GROUP IS {integer-3
PLUS integer-4
NEXT PAGE}]

TYPE IS {
 { REPORT HEADING }
 RH
 { PAGE HEADING }
 PH
 { CONTROL HEADING } { data-name-2 }
 { CH } { FINAL }
 { DETAIL }
 DE
 { CONTROL FOOTING } { data-name-3 }
 { CF } { FINAL }
 { PAGE FOOTING }
 PF
 { REPORT FOOTING }
 RE
 }
 { USAGE IS } DISPLAY }

Format 2

level-number { data-name-1 }
 { LINE NUMBER IS { integer-1 { ON NEXT PAGE } }
 { PLUS integer-2 }
 { USAGE IS } DISPLAY }

Format 3

level-number { data-name-1 }
 { BLANK WHEN ZERO }
 { GROUP INDICATE }
 { { JUSTIFIED } RIGHT }
 { JUST }
 { LINE NUMBER IS { integer-1 { ON NEXT PAGE } }
 { PLUS integer-2 }
 { COLUMN NUMBER IS integer-3 }
 { PICTURE } IS character-string
 { PIC }
 {
 { SOURCE IS identifier-1
 { VALUE IS literal
 { SUM { identifier-2 } ... { UPON { data-name-2 } ... } ...
 { RESET ON { data-name-4 } }
 { FINAL }
 }
 { USAGE IS } DISPLAY }

General Format For Procedure Division

Format 1

PROCEDURE DIVISION { USING { data-name-1 } ... }
DECLARATIVES .
 { section-name SECTION { segment-number } declarative-sentence
 { paragraph-name { sentence } ... } ...
END DECLARATIVES .

{section-name SECTION {segment-number}}
 {paragraph-name {sentence}...}...

Format 2

PROCEDURE DIVISION (USING {data-name-1}...) =
 {paragraph-name {sentence}...}

General Format For Verbs

ACCEPT identifier [FROM mnemonic-name]

ACCEPT identifier FROM {
 DATE
 DAY
 TIME

ACCEPT cd-name MESSAGE COUNT

ADD {identifier-1
 literal-1} ... TO {identifier-m (ROUNDED)} ...

{ON SIZE ERROR imperative-statement}

ADD {identifier-1
 literal-1} {identifier-2
 literal-2} ... GIVING {identifier-m (ROUNDED)} ...

{ON SIZE ERROR imperative-statement}

ADD {CORRECTING
 CORR} identifier-1 TO identifier-2 (ROUNDED)

{ON SIZE ERROR imperative-statement}

ALTER {procedure-name-1 TO (PROCEED TO) procedure-name-2} ...

CALL {identifier-1
 literal-1} {USING {data-name-1}...}

{ON OVERFLOW imperative-statement}

CANCEL {identifier-1
 literal-1} ...

CLOSE {file-name-1 {
 {REEL {WITH NO REWIND}
 UNIT {FOR REMOVAL}
 WITH {NO REWIND}
 LOCK}
 ...}

CLOSE {file-name-1 {WITH LOCK} ...

COMPUTE {identifier-1 (ROUNDED)} ... = arithmetic-expression

{ON SIZE ERROR imperative-statement}

COPY text-name {OF
 IN} library-name

{
 REPLACING {
 = pseudo-text-1 =
 identifier-1
 literal-1
 word-1
 } BY {
 = pseudo-text-2 =
 identifier-2
 literal-2
 word-2
 } ...}

DELETE file-name RECORD {INVALID KEY imperative-statement}

DISABLE {INPUT (TERMINAL)
 OUTPUT} cd-name WITH KEY {identifier-1
 literal-1}

DISPLAY {identifier-1
 literal-1} ... {UPON mnemonic-name}

DIVIDE {identifier-1
literal-1} INTO {identifier-2 [ROUNDED]} ...
 [ON SIZE ERROR imperative-statement]
DIVIDE {identifier-1
literal-1} INTO {identifier-2
literal-2} GIVING {identifier-3 [ROUNDED]} ...
 [ON SIZE ERROR imperative-statement]
DIVIDE {identifier-1
literal-1} BY {identifier-2
literal-2} GIVING {identifier-3 [ROUNDED]} ...
 [ON SIZE ERROR imperative-statement]
DIVIDE {identifier-1
literal-1} INTO {identifier-2
literal-2} GIVING {identifier-3 [ROUNDED]}
REMAINDER {identifier-4 [ON SIZE ERROR imperative-statement]}
DIVIDE {identifier-1
literal-1} BY {identifier-2
literal-2} GIVING {identifier-3 [ROUNDED]}
REMAINDER {identifier-4 [ON SIZE ERROR imperative-statement]}
ENABLE {INPUT [TERMINAL]} col-name WITH KEY {identifier-1
literal-1}
OUTPUT
ENTER language-name {routine-name}
EXIT [PROGRAM]
GENERATE {data-name
report-name}
GO TO {procedure-name-1}
GO TO {procedure-name-1} ... procedure-name-n
DEPENDENT ON identifier
IF condition {statement-1
NEXT SENTENCE} {ELSE statement-2
ELSE NEXT SENTENCE}
INITIATE {report-name-1} ...
INSPECT identifier-1 TALLYING
 { identifier-2 FOR { { ALL
LEADING } { identifier-3
literal-1 } } { { BEFORE
AFTER } INITIAL { identifier-4
literal-2 } } } ... }
CHARACTERS
INSPECT identifier-1 REPLACING
 { CHARACTERS BY { identifier-6
literal-4 } { { BEFORE
AFTER } INITIAL { identifier-7
literal-5 } } }
 { { { ALL
LEADING } { identifier-5
literal-3 } } BY { identifier-6
literal-4 } { { BEFORE
AFTER } INITIAL { identifier-7
literal-5 } } } } ... }
FIRST
INSPECT identifier-1 TALLYING
 { identifier-2 FOR { { { ALL
LEADING } { identifier-3
literal-1 } } } { { BEFORE
AFTER } INITIAL { identifier-4
literal-2 } } } } ... }
CHARACTERS
REPLACING
 { CHARACTERS BY { identifier-6
literal-4 } { { BEFORE
AFTER } INITIAL { identifier-7
literal-5 } } }
 { { { ALL
LEADING } { identifier-5
literal-3 } } BY { identifier-6
literal-4 } { { BEFORE
AFTER } INITIAL { identifier-7
literal-5 } } } } ... }
FIRST

MERGE file-name-1 { ON { ASCENDING
DESCENDING } KEY {data-name-1}... }...

[COLLATING SEQUENCE IS alphabet-name]

USING file-name-2 {file-name-3}...

{ OUTPUT PROCEDURE IS section-name-1 { { THROUGH
THRU } section-name-2 }
GIVING file-name-5 }

MOVE { identifier-1
literal } TO { identifier-2 }...

MOVE { CORRESPONDING
CORR } identifier-1 TO identifier-2

MULTIPLY { identifier-1
literal-1 } BY { identifier-2 { ROUNDED } }...

[ON SIZE ERROR imperative-statement]

MULTIPLY { identifier-1
literal-1 } BY { identifier-2
literal-2 } GIVING { identifier-3 { ROUNDED } }...

[ON SIZE ERROR imperative-statement]

{ INPUT { file-name-1 { REVERSED
WITH NO REWIND } } }...
OPEN { OUTPUT { file-name-3 { WITH NO REWIND } } }...
1-Q { file-name-5 }...
{ EXTEND { file-name-7 } }...

PERFORM procedure-name-1 { { THROUGH
THRU } procedure-name-2 }

PERFORM procedure-name-1 { { THROUGH
THRU } procedure-name-2 } { identifier-1
integer-1 } TIMES

PERFORM procedure-name-1 { { THROUGH
THRU } procedure-name-2 } UNTIL condition-1

PERFORM procedure-name-1 { { THROUGH
THRU } procedure-name-2 }

VARYING { identifier-2
index-name-1 } FROM { identifier-3
index-name-2
literal-1 }

BY { identifier-4
literal-3 } UNTIL condition-1

{ AFTER { identifier-5
index-name-3 } FROM { identifier-6
index-name-4
literal-3 } }

BY { identifier-7
literal-4 } UNTIL condition-2 }

{ AFTER { identifier-8
index-name-5 } FROM { identifier-9
index-name-6
literal-5 } }

$$\left[\begin{array}{l} \text{BY } \left\{ \begin{array}{l} \text{identifier-10} \\ \text{literal-6} \end{array} \right\} \text{ UNTIL } \text{condition-3} \end{array} \right]$$

READ file-name [NEXT] RECORD [INTO identifier] [AT END imperative-statement]
READ file-name RECORD [INTO identifier] [KEY IS data-name]
 [INVALID KEY imperative-statement]

RECEIVE cd-name $\left\{ \begin{array}{l} \text{MESSAGE} \\ \text{SEGMENT} \end{array} \right\}$ INTO identifier-1 [NO DATA imperative-statement]

MULTIPLY $\left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\}$ BY $\left\{ \begin{array}{l} \text{identifier-2} \\ \text{literal-2} \end{array} \right\}$ GIVING [identifier-3 [ROUNDED]] ...
 [ON SIZE ERROR imperative-statement]

$$\left[\begin{array}{l} \text{OPEN } \left\{ \begin{array}{l} \text{INPUT } \left\{ \begin{array}{l} \text{file-name-1} \\ \text{WITH NO REWIND} \end{array} \right\} \dots \\ \text{OUTPUT } \left\{ \begin{array}{l} \text{file-name-3} \\ \text{WITH NO REWIND} \end{array} \right\} \dots \\ \text{I-O } \left\{ \begin{array}{l} \text{file-name-5} \\ \dots \end{array} \right\} \\ \text{EXTEND } \left\{ \begin{array}{l} \text{file-name-7} \\ \dots \end{array} \right\} \end{array} \right\} \end{array} \right]$$

PERFORM procedure-name-1 $\left\{ \begin{array}{l} \text{THROUGH} \\ \text{THRU} \end{array} \right\}$ procedure-name-2

PERFORM procedure-name-1 $\left\{ \begin{array}{l} \text{THROUGH} \\ \text{THRU} \end{array} \right\}$ procedure-name-2 $\left\{ \begin{array}{l} \text{identifier-1} \\ \text{integer-1} \end{array} \right\}$ TIMES

PERFORM procedure-name-1 $\left\{ \begin{array}{l} \text{THROUGH} \\ \text{THRU} \end{array} \right\}$ procedure-name-2 UNTIL condition-1

PERFORM procedure-name-1 $\left\{ \begin{array}{l} \text{THROUGH} \\ \text{THRU} \end{array} \right\}$ procedure-name-2

VARYING $\left\{ \begin{array}{l} \text{identifier-2} \\ \text{index-name-1} \end{array} \right\}$ FROM $\left\{ \begin{array}{l} \text{identifier-3} \\ \text{index-name-2} \\ \text{literal-1} \end{array} \right\}$

$$\left[\begin{array}{l} \text{BY } \left\{ \begin{array}{l} \text{identifier-4} \\ \text{literal-3} \end{array} \right\} \text{ UNTIL } \text{condition-1} \\ \left[\begin{array}{l} \text{AFTER } \left\{ \begin{array}{l} \text{identifier-5} \\ \text{index-name-3} \end{array} \right\} \text{ FROM } \left\{ \begin{array}{l} \text{identifier-6} \\ \text{index-name-4} \\ \text{literal-3} \end{array} \right\} \\ \text{BY } \left\{ \begin{array}{l} \text{identifier-7} \\ \text{literal-4} \end{array} \right\} \text{ UNTIL } \text{condition-2} \end{array} \right]$$

$$\left[\begin{array}{l} \text{AFTER } \left\{ \begin{array}{l} \text{identifier-8} \\ \text{index-name-5} \end{array} \right\} \text{ FROM } \left\{ \begin{array}{l} \text{identifier-9} \\ \text{index-name-6} \\ \text{literal-5} \end{array} \right\} \\ \text{BY } \left\{ \begin{array}{l} \text{identifier-10} \\ \text{literal-6} \end{array} \right\} \text{ UNTIL } \text{condition-3} \end{array} \right]$$

READ file-name [NEXT] RECORD [INTO identifier] [AT END imperative-statement]
READ file-name RECORD [INTO identifier] [KEY IS data-name]
 [INVALID KEY imperative-statement]

RECEIVE cd-name { MESSAGE } INTO identifier-1 { NO DATA imperative-statement }
SEGMENT

RELEASE record-name { FROM identifier }

RETURN file-name RECORD { INTO identifier } AT END imperative-statement

REWRITE record-name { FROM identifier } INVALID KEY imperative-statement

SEARCH identifier-1 { VARYING { identifier-2
index-name-1 } } { AT END imperative-statement-1 }
{ WHEN condition-1 { imperative-statement-2 } } ...
NEXT SENTENCE

SEARCH ALL identifier-1 { AT END imperative-statement-1 }

WHEN { data-name-1 { IS EQUAL TO { identifier-3
literal-1
arithmetic-expression-1 } }
condition-name-1 }
{ AND { data-name-2 { IS EQUAL TO { identifier-4
literal-2
arithmetic-expression-2 } } } ...
condition-name-2 }
{ imperative-statement-2 }
NEXT SENTENCE

SEND cd-name FROM identifier-1

SEND cd-name { FROM identifier-1 } { WITH identifier-2 }
{ WITH ESI }
{ WITH EMI }
{ WITH EGI }

{ BEFORE } ADVANCING { { identifier-3 { LINE } } }
{ AFTER } { { integer { LINES } } }
{ { mnemonic-name } }
{ PAGE }

SET { identifier-1 } ... TO { identifier-3
index-name-1 } { index-name-3
integer-1 }

SET { index-name-4 } ... { UP BY } { identifier-4
DOWN BY } { integer-2 }

SORT file-name-1 { ON { ASCENDING
DESCENDING } KEY { data-name-1 } ... } ...
{ COLLATING SEQUENCE IS alphabet-name }

{ INPUT PROCEDURE IS section-name-1 { { THROUGH
THRU } section-name-2 } }
{ USING (file-name-2) ... }

{ OUTPUT PROCEDURE IS section-name-3 { { THROUGH
THRU } section-name-4 } }
{ GIVING file-name-4 }

START file-name $\left[\begin{array}{c} \text{KEY} \left\{ \begin{array}{l} \text{IS } \underline{\text{EQUAL TO}} \\ \text{IS } = \\ \text{IS } \underline{\text{GREATER THAN}} \\ \text{IS } > \\ \text{IS } \underline{\text{NOT LESS THAN}} \\ \text{IS } \underline{\text{NOT}} < \end{array} \right\} \text{data-name} \end{array} \right]$

[INVALID KEY imperative-statement]

STOP $\left\{ \begin{array}{l} \underline{\text{RUN}} \\ \text{literal} \end{array} \right\}$

STRING $\left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \dots \underline{\text{DELIMITED BY}} \left\{ \begin{array}{l} \text{identifier-3} \\ \text{literal-3} \\ \underline{\text{SIZE}} \end{array} \right\} \dots$

INTO identifier-7 [WITH POINTER identifier-8]

[ON OVERFLOW imperative-statement]

SUBTRACT $\left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \dots \underline{\text{FROM}} \left\{ \begin{array}{l} \text{identifier-m} \\ \text{literal-m} \end{array} \right\} \underline{\text{(ROUNDED)}} \dots$

[ON SIZE ERROR imperative-statement]

SUBTRACT $\left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \dots \underline{\text{FROM}} \left\{ \begin{array}{l} \text{identifier-m} \\ \text{literal-m} \end{array} \right\}$

GIVING (identifier-n (ROUNDED))...

[ON SIZE ERROR imperative-statement]

SUBTRACT (CORRESPONDING) identifier-1 FROM identifier-2 (ROUNDED)

[ON SIZE ERROR imperative-statement]

SUPPRESS PRINTING

TERMINATE (report-name-1)...

UNSTRING identifier-1

$\left[\underline{\text{DELIMITED BY}} \left\{ \begin{array}{l} \text{identifier-2} \\ \text{literal-1} \end{array} \right\} \left\{ \begin{array}{l} \underline{\text{OR}} \left\{ \begin{array}{l} \text{identifier-3} \\ \text{literal-2} \end{array} \right\} \dots \right\} \right]$

INTO identifier-4 (DELIMITER IN identifier-5 [COUNT IN identifier-6])...

[WITH POINTER identifier-10 [TALLYING IN identifier-11]

[ON OVERFLOW imperative-statement]

USE AFTER STANDARD $\left\{ \begin{array}{l} \underline{\text{EXCEPTION}} \\ \underline{\text{ERROR}} \end{array} \right\} \underline{\text{PROCEDURE ON}} \left\{ \begin{array}{l} \text{(file-name-1) ...} \\ \underline{\text{INPUT}} \\ \underline{\text{OUTPUT}} \\ \underline{\text{I-O}} \\ \underline{\text{EXTEND}} \end{array} \right\}.$

USE BEFORE REPORTING identifier-

USE FOR DEBUGGING ON $\left\{ \begin{array}{l} \text{cd-name-1} \\ \left\{ \begin{array}{l} \underline{\text{ALL REFERENCES OF}} \text{ identifier-1} \\ \text{file-name-1} \\ \text{procedure-name-1} \\ \underline{\text{ALL PROCEDURES}} \end{array} \right\} \end{array} \right\} \dots$

WRITE record-name [FROM identifier-1]

$\left\{ \begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right\} \underline{\text{ADVANCING}} \left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{identifier-2} \\ \text{integer} \end{array} \right\} \left\{ \begin{array}{l} \underline{\text{LINE}} \\ \underline{\text{LINES}} \end{array} \right\} \\ \left\{ \begin{array}{l} \text{mnemonic-name} \\ \underline{\text{PAGE}} \end{array} \right\} \end{array} \right\}$

$$\left[\text{AT } \left\{ \begin{array}{l} \text{END-OF-PAGE} \\ \text{EOP} \end{array} \right\} \text{imperative-statement} \right]$$

$$\text{WRITE record-name } \left[\text{FROM identifier} \right] \text{INVALID KEY imperative-statement}]$$

General Format For Conditions

Relation Condition

$$\left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \\ \text{arithmetic-expression-1} \\ \text{index-name-1} \end{array} \right\} \left\{ \begin{array}{l} \text{IS } \left(\text{NOT} \right) \text{ GREATER THAN} \\ \text{IS } \left(\text{NOT} \right) \text{ LESS THAN} \\ \text{IS } \left(\text{NOT} \right) \text{ EQUAL TO} \\ \text{IS } \left(\text{NOT} \right) > \\ \text{IS } \left(\text{NOT} \right) < \\ \text{IS } \left(\text{NOT} \right) = \end{array} \right\} \left\{ \begin{array}{l} \text{identifier-2} \\ \text{literal-2} \\ \text{arithmetic-expression-2} \\ \text{index-name-2} \end{array} \right\}$$

Class condition

$$\text{identifier IS } \left(\text{NOT} \right) \left\{ \begin{array}{l} \text{NUMERIC} \\ \text{ALPHABETIC} \end{array} \right\}$$

Sign condition

$$\text{arithmetic-expression } \left(\text{NOT} \right) \left\{ \begin{array}{l} \text{POSITIVE} \\ \text{NEGATIVE} \\ \text{ZERO} \end{array} \right\}$$

Condition-name condition

condition-name

Switch-status condition

condition-name

Negated simple condition

NOT simple-condition

Combined condition

$$\text{condition } \left\{ \left\{ \begin{array}{l} \text{AND} \\ \text{OR} \end{array} \right\} \text{condition} \right\} \dots$$

Abbreviated combined relation condition

$$\text{relation-condition } \left\{ \left\{ \begin{array}{l} \text{AND} \\ \text{OR} \end{array} \right\} \left(\text{NOT} \right) \left[\text{relational-operator} \right] \text{object} \right\} \dots$$

Miscellaneous Formats

Qualification

$$\left\{ \begin{array}{l} \text{data-name-1} \\ \text{condition-name} \end{array} \right\} \left\{ \left\{ \begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right\} \text{data-name-2} \right\} \dots$$

$$\text{paragraph-name } \left\{ \left\{ \begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right\} \text{section-name} \right\}$$

$$\text{text-name } \left\{ \left\{ \begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right\} \text{library-name} \right\}$$

Subscripting

$$\left\{ \begin{array}{l} \text{data-name} \\ \text{condition-name} \end{array} \right\} (\text{subscript-1 } [, \text{subscript-2 } [, \text{subscript-3}]])$$

Indexing

$$\left\{ \begin{array}{l} \text{data-name} \\ \text{condition-name} \end{array} \right\} \left\{ \left\{ \begin{array}{l} \text{index-name-1} \\ \text{literal-1} \end{array} \right\} \left[\begin{array}{l} \text{literal-2} \\ \text{literal-3} \end{array} \right] \left[\begin{array}{l} \text{index-name-2} \\ \text{literal-4} \end{array} \right] \left[\begin{array}{l} \text{index-name-3} \\ \text{literal-5} \end{array} \right] \left[\begin{array}{l} \text{literal-6} \end{array} \right] \right\} \right\}$$
Identifier Format 1
$$\text{data-name-1} \left[\left\{ \frac{\text{OF}}{\text{IN}} \right\} \text{data-name-2} \right] \dots ((\text{subscript-1} [, \text{subscript-2} [, \text{subscript-3}]])$$
Identifier Format 2[illegible]

١٥-٤ معجم لأهم المصطلحات المستخدمة في لغة كوبول :

طريقة المعالجة ACCESS MODE

وهي الطريقة التي تعالج بها السجلات ضمن الملف

ACTUAL DECIMAL POINT الفاصلة العشرية الفعلية

التمثيل الفعل للفاصلة العشرية في الحقل وقد تكون اشارة (.) أو (,).

ALPHABETIC CHARACTERS الرمز الهجائي

وهو الرمز الذي يدل على أحد الحروف في المجموعة التالية :

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q,

R, S, T, U, V, W, X, Y, Z, SPACE والفراغ

الرمز الهجائي / الرقمي المختلط ALPHANUMERIC CHARACTER

عبارة عن أي رمز يستخدم في مجموعة رموز الحاسب.

• **Introductory Structured Cobol Programming.** By Gray S. Ponkin.

ARITHMATIC EXPRESSION (حسابي) تعبير رياضي

يمكن أن يكون التعبير الحسابي حقلاً رقمياً أو ثابتاً رقمياً أو مجموعة من كليهما تفصل بينهما إشارة عملية حسابية أو يحيط بهما أقواس .

ARITHMATIC OPERATOR (رياضي) عامل حسابي

هو رمز (أو رمزان متجانسان) يدل على ما يلي :

| الرمز | المعنى |
|-------|----------------------|
| + | إضافة |
| - | طرح |
| * | ضرب |
| / | قسمة |
| ** | الرفع إلى قوة (أس) . |

ASCENDING KEY المفتاح التصاعدي

وهو حقل يحتوي على قيمة ترتب وفقها البيانات بصورة تصاعدية ، تبدأ بالقيمة الأقل وتنتهي بالقيمة الأعلى .

ASSUMED DECIMAL POINT الفاصلة العشرية الافتراضية

وهي المكان الذي يفترض أن توجد فيه الفاصلة العشرية إلا أنها لا تمثل برمز فعلياً .

AT END CONDITION شرط النهاية

يكتب هذا الشرط في الحالات التالية :

- ١ - أثناء تنفيذ تعليمة القراءة READ لملف معالج بالطريقة التسلسلية .
- ٢ - أثناء تنفيذ تعليمة العودة RETURN عندما لا يوجد سجل تالي للمعالجة

بالنسبة للملفات الدمج والفرز SORT AND MERGE

٣- أثناء تنفيذ تعليمة البحث SEARCH عندما تنتهي عمليات البحث دون وجود القيمة المحددة بعبارة WHEN

كتلة بيانات BLOCK

وهي مجموعة من السجلات ، وكل مجموعة تتألف من سجل واحد أو أكثر .

رمز CHARACTER

وهو أصغر وحدة أساسية من اللغة .

سلسلة الرموز CHARACTER STRING

وهي عبارة عن سلسلة متصلة من الرموز تشكل إما كلمة كويول أو ثابتاً حرفياً ، أو شكل PICTURE أو ملاحظة .

شرط النوع CLASS CONDITION

وهو الافتراض الذي من أجله يمكن للقيمة الحقيقية أن تقرر بأن قيمة حقل معين بكامله إما ابجدياً أو رقمياً .

مجموعة رموز لغة كويول COBOL CHARACTER SET

إن مجموعة الرموز المستخدمة في لغة كويول هي ٥١ (واحد وخمسين) رمزاً مصنفة

المعنى

الرمز

كالتالي :

الأرقام من الصفر إلى التسعة

الأرقام : من الصفر إلى 9

الأحرف الهجائية من A إلى Z

A → Z

| | |
|--------------------------|----|
| الفراغ (ويمثل بحرف واحد) | |
| إشارة الجمع | + |
| إشارة الناقص (أو الشرطة) | - |
| إشارة النجمة | * |
| إشارة الخط المائل | / |
| إشارة المساواة | = |
| إشارة الدولار | \$ |
| الفاصلة | , |
| الفاصلة المنقوطة | ; |
| النقطة (الفاصلة العشرية) | . |
| علامة الاقتباس | » |
| القوس الأيسر | (|
| القوس الأيمن |) |
| إشارة أكبر من | > |
| إشارة أصغر من | < |

عمود (موقع) COLUMN

هو موقع الرمز في سطر الطباعة، و يرقم ابتداءً من الرقم ١ أقصى يسار السطر ولغاية الجهة اليمنى نهاية السطر.

الشرط المركب COMBINED CONDITION

وهو حاصل ربط شرطين أو أكثر بعملية AND أو OR المنطقية.

سطر الملاحظات COMMENT LINE

يوجد هذا السطر في برنامج كويول مكتوبا بصيغة المصدر (الأصل) SOURCE
و يعرف بوضع علامة * في العمود رقم سبعة.
و يوضع هذا السطر من أجل شرح العمليات في البرنامج بصورة موجزة ويخدم في
عملية توثيق البرنامج .

اسم الشرط CONDITION NAME

وهو اسم لحقل يحدد فيه المبرمج قيمة معينة أو مجموعة قيم أو مدى من القيم يمكن
للمتغير الشرطي أن يأخذها .

التعبير الشرطي CONDITIONAL EXPRESSION

وهو شرط بسيط أو شرط مركب يعين بتعليمة IF أو PERFORM أو SEARCH

فصل مكونات الحاسب CONFIGURATION SECTION

وهو جزء من قسم المحيطات ENVIRONMENT DIVISION الذي يبين
مواصفات الحاسب المستخدم .

الحقول المتتابة CONTIGUOUS ITEMS (FIELDS)

وهي الحقول التي توصف بشكل مداخل متعاقبة في قسم البيانات
DATA DIVISION وتظهر ذات علاقة هرمية مع بعضها البعض .

CONTROL BREAK

وهو التغير الذي يحصل على قيمة حقل معتبر كحقل تحكم في عبارة CONTROL

COUNTER العداد

عبارة عن حقل يستخدم لتخزين الأرقام التي تمثل بطريقة تسمح لهذه الأرقام بالزيادة أو النقصان بواسطة قيمة أخرى. أو تبديل هذه الأرقام أو إعادتها للقيمة صفر أو لأي قيمة اعتباطية موجبة أو سالبة.

CURRENT RECORD السجل الحالي

وهو السجل الذي يوجد في المنطقة المخصصة للملف في الذاكرة من أجل إجراء العمليات اللازمة عليه.

DATA ITEM عنصر البيانات

وهو رمز أو مجموعة رموز متصلة يمكن تعريفها على أنها وحدة من البيانات في برنامج كوبرل.

DATA NAME اسم البيان (حقل)

عبارة عن كلمة تحدد من قبل المبرمج لتسمية عنصر البيانات الموصوف في قسم

DATA DIVISION البيانات

DELIMITER رمز النهاية

رمز أو سلسلة رموز متصلة التي يمكن أن تحدد نهاية سلسلة من الرمز وتفصلها عن سلسلة أخرى تالية.

DESCENDING KEY المفتاح التنازلي

وهو حقل أو مفتاح يحتوي على قيمة ترتب وفقها البيانات بصورة تنازلية تبدأ

بالقيمة الأعلى وتنتهي بالقيمة الأقل .

قسم (من برنامج كوبول) DIVISION

يتكون من فصل واحد أو أكثر وكل فصل يتكون من مجموعة فقرات يدعى هيكل القسم ويمكن ألا يحتوي القسم على أي فصل .
وهناك ٤ أقسام في برنامج كوبول وهي :

قسم التعريف IDENTIFICATION DIVISION

قسم المحيطات ENVIRONMENT DIVISION

قسم البيانات DATA DIVISION

قسم الاجراءات PROCEDURE DIVISION

رمز الاخراج EDITING CHARACTER

وهو عبارة عن رمز أو رمزين ثابتين يشكلان ما يلي :

| الرمز | المعنى |
|-------|--|
| B | فراغ |
| 0 | صفر |
| + | الاشارة الجبرية الموجبة |
| — | الاشارة الجبرية السالبة |
| CR | إشارة دائن |
| DB | إشارة مئين |
| Z | طمس الصفر (تمثيل الصفر على يسار الرقم بفراغ) |
| \$ | إشارة الدولار |
| , | الفاصلة (الفاصلة العشرية) |

النقطة (الفاصلة العشرية)

الخط المائل

ELEMENTARY ITEM العنصر الأول

وهو عنصر البيانات الموصوف بشكل لا يمكن أن يجزأ من الناحية النظرية.

ENTRY الجملة المستقلة

عبارة عن مجموعة متعاقبة من العبارات الوصفية تنتهي بنقطة وتوجد في أقسام التعريف والمخططات والبيانات في برنامج كوبرول.

FIGURATIVE CONSTANT الثابت الشكلي

وهي قيم تولد عن المترجم COMPILER وتعتبر ككلمات محجوزة (مثل SPACE و ZERO ...).

FILE الملف

وهو مجموعة من السجلات.

FILE ORGANIZATION تنظيم الملف

وهي البيئة النظرية الثابتة التي أعطيت للملف عند إنشائه.

FILE SECTION فصل توصيف الملفات

عبارة عن فصل في قسم البيانات ويحتوي على جل وصفية والملف مع توصيفات السجل الخاصة بالملف.

شكل ، صيغة FORMAT

وهي ترتيب محدد لمجموعة من البيانات .

مجموع عناصر بيانات GROUP ITEM

وهي مجموعة متصلة من عناصر أولية أو من عناصر مجمعة للبيانات وتعطى اسماً واحداً .

النهاية الاعظمية HIGH ORDER END

وهو الحرف الموجود في أقصى يسار سلسلة رموز.

طريقة إدخال - إخراج I-O MODE

وهي حالة الملف بعد تنفيذ تعليمة فتح الملف OPEN بعبارة I-O (إدخال - إخراج) وقبل إغلاق الملف بتعليمة CLOSE

اسم حقل (بيان) IDENTIFIER

اسم لبيان يتبع لقواعد لغة كوبرول من حيث تشكيله ، ويمكن أن يكون مركباً من أسماء معرفة وأدلة ومؤشرات لازمة لجعله مرجحاً وحيداً لعنصر البيانات في برنامج كوبرول .

التعليمة الأمرية IMPERATIVE STATEMENT

وهي التعليمة التي تبدأ بفعل أمر وتعين عملاً غير مشروط يجب أن ينفذ ويمكن للتعليمة الأمرة أن تحتوي على سلسلة من التعليمات الأمرة الأخرى .

الفهرس (المؤشر) INDEX

وهو موقع محدد في ذاكرة الحاسب، وتمثل محتوياته تعريف لعنصر محدد في جدول

TABLE

اسم الفهرس (المؤشر) INDEX-NAME

كلمة تعرف من قبل المبرمج لتسمية مؤشر أو فهرس لجدول محدد.

الملف المفهرس INDEX FILE

وهو الملف ذو التنظيم المفهرس.

التنظيم المفهرس INDEXED ORGANIZATION

وهو الهيكل النظري الثابت للملف، حيث يعرف كل سجل فيه بقيمة مفتاح واحد أو أكثر ضمن سجل هذا الملف.

ملف الإدخال INPUT FILE

هو الملف المفتوح للمعالجة بطريقة الإدخال (و يستخدم للقراءة فقط).

صيغة الإدخال INPUT MODE

وهي حالة الملف بعد تنفيذ تعليمة OPEN بعبارة INPUT وقبل تنفيذ تعليمة إغلاق الملف CLOSE

ملف إدخال - إخراج INPUT-OUTPUT FILE

وهو الملف المفتوح للمعالجة بصيغة إدخال - إخراج (ويمكن القراءة من هذا الملف

والكتابة عليه).

فصل الإدخال - الإخراج INPUT-OUTPUT SECTION

وهو فصل من قسم المحيطات ENVIRONMENT DIVISION في برنامج كوبرول ويستخدم لتسمية الملفات والوسائط الخارجية اللازمة لاستخدامات البرنامج والتي تزود البرنامج بالمعلومات المطلوبة للمعالجة أثناء تنفيذ البرنامج .

العدد الصحيح INTEGER

وهو الحقل الرقمي أو العدد الذي لا يحتوي في الجزء الأيمن من الفاصلة العشرية المقترضة على أي رقم .

شرط عدم صلاحية المفتاح INVALID KEY CONDITION

يتحقق هذا الشرط عندما تكون القيمة المحددة للمفتاح غير مطابقة لمحتويات الفهرس في الملفات النسبية والمفهرسة .

المفتاح KEY

وهو الحقل الذي يعرف موقعاً من السجل ، أو مجموعة الحقول التي تحدد تسلسل البيانات .

رقم المستوى LEVEL NUMBER

وهو رقم محدد من قبل المبرمج ويشير إلى موقع عنصر البيانات من البنية الهرمية للسجل ، أو هو الرقم الذي يشير إلى مميزات خاصة لهذا العنصر . ويمثل رقم المستوى بخانة واحدة أو خانتين رقميتين وتتراوح قيمتها من ١ إلى ٩٩ مشيراً بذلك إلى البنية

الهرمية لعنصر السجل . والأرقام من ١ — ٩ قد تمثل مفردة أو قد يسبقها صفر.

حرفي (قيمة ثابتة) LITERAL

هي سلسلة الرموز التي تكون قيمتها هي نفس الرموز المكونة لها ولا تحتاج إلى اسم متغير.

العامل المنطقي LOGICAL OPERATOR

هو أحد الكلمات المحجوزة التالية AND أو OR أو NOT المصاغ في شكل شرط ، إن كلاً من أو إحدى الكلمتين OR و AND تستخدم بعمليات الربط المنطقية ، أما كلمة NOT فتستخدم من أجل النفي المنطقي .

السجل المنطقي LOGICAL RECORD

وهو الاسم الشامل لمجموعة عناصر بيانات و يشار إليه برقم مستوى 01 دوماً .

النهاية الدنيا LOW ORDER END

وهي الحرف الموجود في أقصى يمين سلسلة من الرموز.

تخزين كتل البيانات MASS STORAGE

وهو وسط التخزين الذي يمكن للبيانات أن تنظم أو تعالج عليه بطريقة تسلسلية وغير تسلسلية .

ملف تخزين كتل البيانات MASS STORAGE FILE

وهو مجموعة من السجلات موجودة على وسط تخزين خاص بكتل البيانات .

الاسم الرمزي (المنطقي) لوسط تخزين MNEMONIC NAME
هي الكلمة التي يحددها المبرمج في قسم المحيطات في برنامج كويول وتشير إلى
وسط تخزين محدد.

الشرط المركب المنفي NEGATED COMBINED CONDITION
وهو عامل النفي NOT متبوعاً بشرط مركب محصور ضمن قوسين.

الشرط البسيط المنفي NEGATED SIMPLE CONDITION
وهو عامل النفي NOT متبوعاً بشرط بسيط.

الجملة المنفذة التالية NEXT EXECUTABLE SENTENCE
وهي الجملة التالية التي سينتقل إليها تسلسل تنفيذ البرنامج بعد أن ينتهي من
تنفيذ الجملة الحالية.

عناصر البيانات المستقلة NON CONTIGUOUS ITEMS
وهي عناصر أولية للبيانات، غير قابلة للتجزئة موصفة في فصل التخزين الداخلي
WORKING-STORAGE SECTION والتي ليس لها علاقات هرمية مع عناصر
أخرى، وتأخذ رقم المستوى 77 أو 01.

عنصر البيانات غير الرقمي NON NUMERIC ITEM
وهو العنصر الموصوف بشكل يسمح بأن يحتوي على أي تركيبة من الرموز المصنفة
ضمن مجموعة الرموز المستخدمة في الحاسب الآلي.

السلسلة غير الرقمية (الثابت غير الرقمي) NON NUMERIC LITERAL

وهي سلسلة من الرموز محددة بعلامتي اقتباس " ويمكن أن تحتوي سلسلة الرموز هذه على أي رمز من مجموعة الرموز المستخدمة في الحاسب الآلي .

الرمز الرقمي NUMERIC CHARACTER

وهي أي رمز يخص مجموعة الرموز التالية :

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

عنصر البيانات الرقمي NUMERIC ITEM

وهو عنصر البيانات الموصف بشكل يقتصر محتواه على قيم رقمية فقط أي من الصفر إلى التسعة ، وإذا كان موصفاً بإشارة فيمكن أن يحتوي على إشارة جبرية موجبة أو سالبة .

رموز رقمية (ثابت رقمي) NUMERIC LITERAL

وهي سلسلة رموز تتألف من رقم واحد أو أكثر أو يمكن أن تكون السلسلة عشرية أو تحتوي على إشارة جبرية موجبة أو سالبة أو قد تحتوي على فاصلة عشرية مع إشارة جبرية ، والفاصلة العشرية يجب ألا تكون أول حرف من اليمين (بل ضمن السلسلة) أما الإشارة الجبرية إن وجدت فانها يجب أن تكون بأقصى يسار السلسلة .

البرنامج بلغة الآلة OBJECT PROGRAM

وهو مجموعة من أوامر التنفيذ الخاصة بلغة الآلة مع أشياء أخرى مصممة لأن تكون الوسيط في معالجة البيانات ، وتشير عبارة OBJECT PROGRAM بصورة عامة إلى ناتج عملية الترجمة COMPILE لبرنامج مكتوب بلغة كوبول .

صيغة فتح ملف OPEN MODE

وهي حالة الملف بعد تنفيذ تعليمة فتح الملف OPEN وقبل تنفيذ تعليمة إغلاق CLOSE الملف .

وتحدد صيغة الفتح مع تعليمة OPEN بأحد الأشكال التالية :

يستخدم عندها الملف كادخال INPUT

يستخدم عندها الملف كإخراج OUTPUT

يستخدم الملف كادخال - إخراج I-O

يستخدم الملف كإخراج EXTEND

عامل البيانات OPERAND

إن التعريف العام لعبارة عامل البيانات هي أنها أي عنصر يتم العمل عليه .

ومن أجل أغراض الحاسب الآلي تعتبر أي كلمة ترد في التعليمة تمثل اسماً لمخزن

البيانات هي عامل البيانات OPERAND

الاشارة الجبرية (العملية) OPERATIONAL SIGN

وهي إشارة جبرية ملحقه مع حقل رقمي أو سلسلة رقمية لتدل على هذه القيمة إما

موجبة أو سالبة .

ملف إخراج OUTPUT FILE

وهو الملف المفتوح بصيغة الإخراج (للكتابه عليه) .

صيغة الإخراج OUTPUT MODE

وهي حالة الملف بعد تنفيذ تعليمة OPEN بكلمة OUTPUT وقبل تنفيذ تعليمة

إغلاق الملف CLOSE

صفحة PAGE

وهي القسم العمودي من التقرير الذي يمثل البيانات والحقول المفصلة عملياً
بفراغات مناسبة.

جسم الصفحة PAGE BODY

وهو الجزء النظري من الصفحة الذي تكتب فيه الأسطر.

أسفل الصفحة PAGE FOOTING

وهو الجزء السفلي من الصفحة والذي يمثل نهاية الصفحة ويرد في نظام منشيء
التقارير.

رأس الصفحة PAGE HEADING

وهو الجزء العلوي الذي يمثل بداية الصفحة ويحدد بنظام منشيء التقارير.

الفقرة PARAGRAPH

تمثل الفقرة مجموعة تعليمات في قسم الاجراءات تنتهي بنقطة يليها فراغ ويمكن أن
يخصص لها أسم وهذا الأسم ينتهي بنقطة أيضاً أما في قسم التعريف أو المحيطيات
فتمثل الفقرة جملة مستقلة واحدة أو أكثر وتسبق باسم لها وتنتهي بنقطة.

مفتاح سجل رئيسي PRIME RECORD KEY

وهو المفتاح الذي يحتوي على قيمة وحيدة تُعرّف السجل ضمن الملف الرئيسي.

اسم الاجراء PROCEDURE NAME

وهو الاسم الذي يختاره المبرمج ليكون اسماً للفقرة، أو للفصل في قسم
الاجراءات، ويمكن لهذا الاسم أن يكون معرّفاً. ٢

اسم برنامج PROGRAM NAME

وهي الكلمة التي يختارها المبرمج لتكون اسماً لبرنامج المكتوب بلغة كوبرول.

رموز التنقيط PUNCTUATION CHARACTERS

وهي :

| الرمز | المعنى |
|-------|------------------|
| , | الفاصلة |
| ; | الفاصلة المنقوطة |
| . | النقطة |
| » | علامة الاقتباس |
| (| قوس أيسر |
|) | قوس أيمن |
| | فراغ |
| = | علامة المساواة |

اسم عنصر البيانات المكرر QUALIFIED DATA-NAME

وهو اسم عنصر البيانات المؤلف من اسم حقل يليه أحد كلمتي OF أو IN للدلالة
على تبعيته لحقل إجمالي آخر (المرجع).

اسم المرجع (المتحول) QUALIFIER

١ — هو اسم حقل يستخدم كمرجع لاسم حقل آخر مكرر.

٢ — هو اسم فصل SECTION يستخدم كمرجع لاسم فقرة موجودة ضمن هذا الفصل .

طريقة الوصول العشوائية RANDOM ACCESS

وهي الطريقة التي يحدد فيها المبرمج قيمة مفتاح السجل الذي سيعالج في ملف نسبي أو مفهرس ضمن البرنامج .

السجل RECORD

ارجع إلى كلمة LOGICAL RECORD

منطقة السجل RECORD AREA

وهي منطقة مخصصة في الذاكرة الداخلية للحاسب، وذلك لمعالجة السجل الموصف في FD في قسم الملفات FILE SECTION

جولة توصيف ملف RECORD DESCRIPTION ENTRY

هي مجموعة عبارات لتوصيف بيانات سجل معين .

مفتاح السجل RECORD KEY

وهو الحقل الذي بقيمته يعرف السجل في الملف المفهرس .

اسم السجل RECORD NAME

وهي الكلمة التي يختارها المبرمج لتكون اسماً للسجل الموصف في
(FD) FILE DESCRIPTION .

رموز تحديد العلاقة RELATIVE CONDITION

وهو الافتراض الذي تحدد فيه القيمة الحقيقية بأن قيمة التعبير الرياضي أو عنصر
البيانات له علاقة محددة مع قيمة لتعبير رياضي آخر أو لعنصر بيانات آخر.

عامل العلاقة RELATIONAL OPERATION

وهي كلمة محجوزة أو رمز تحديد علاقة أو مجموعة من الكلمات المحجوزة المتتابعة أو
مجموعة من رموز تحديد العلاقة والكلمات المحجوزة المتتابعة معاً تستخدم كوصف
للشرط وهي :

| الرمز | المعنى |
|-----------------------|-------------|
| IS {NOT} GREATER THAN | أكبر من أو |
| IS {NOT} > | ليس أكبر من |
| IS {NOT} LESS THAN | أصغر من أو |
| IS {NOT} < THAN | ليس أصغر من |
| IS {NOT} EQUAL TO | مساوٍ أو |
| IS {NOT} = | غير مساوٍ |

الملف النسبي RELATIVE FILE

هو الملف ذو التنظيم النسبي .

التنظيم النسبي RELATIVE ORGANIZATION

وهو الملف المبني أساساً بحيث أن كل سجل فيه معرف بصورة وحيدة برقم صحيح أكبر من الصفر ويدل على تسلسل هذا السجل في الملف.

جملة توصيف تقرير REPORT DESCRIPTION ENTRY

وهي جملة مستقلة في فصل التقرير REPORT SECTION من قسم البيانات DATA DIVISION مؤلفة من مؤشر مستوى هو RD يليه اسم التقرير يليه مجموعة من عبارات التقرير.

ملف تقرير REPORT FILE

وهو ملف إخراج والذي توصيفه في قسم البيانات يحتوي على عبارة REPORT ويحتوي ملف التقرير هذا هو سجلات مكتوبة بإشراف نظام منشيء التقارير

REPORT WRITTER CONTROL SYSTEM

الكلمة المحجوزة RESERVED WORD

وهي كلمة في لغة كوبول مدرجة ضمن جدول للكلمات تستخدم في برنامج كوبول ولكن يجب ألا تظهر في البرنامج كأسم مستخدم من قبل المبرمج.

فصل SECTION

وهو مجموعة من الفقرات أو الجمل المستقلة يسبقها اسم يعتبر عنواناً لها تدعى فصل ويمكن أن يرد عنوان الفصل دون أن يشمل على فقرات أو جمل.

جملة SENTENCE

وهي عبارة عن تعليمة واحدة أو أكثر تنتهي بنقطة ويلها فراغ واحد على الأقل.

فاصل SEPARATOR

وهو أحد رموز التنقيط و يستخدم للدلالة على انتهاء سلسلة من الرموز.

طريقة الوصول التسلسلية SEQUENTIAL ACCESS

وهي طريقة المعالجة التي يتم فيها استدعاء السجلات من الملف أو وضع السجلات ضمن الملف بشكل متعاقب حسب تسلسلها في الملف.

الملف المتسلسل SEQUENTIAL FILE

وهو الملف المنظم بطريقة تسلسلية.

تنظيم تسلسلي SEQUENTIAL ORGANIZATION

هي بُنية الملف التي تعتمد بأن كل سجل فيه يعرف بعلاقته إلى سابعه وتاليه من السجلات الموجودة في الملف.

شرط الإشارة SIGN CONDITION

وهو الافتراض الذي تقرر فيه القيمة الجبرية للحقل الرقمي أو للتعبير الرياضي، وهل هو أصغر أو أكبر أو مساو للصفر.

ملف فرز SORT FILE

وهو مجموعة من السجلات المراد فرزها بتعليمة فرز SORT و ينشأ ملف الفرز باستخدام تعليمات الفرز فقط، و يعتبر كمنطقة عمل أو تخزين مؤقتة للبيانات المطلوب فرزها.

جملـة توصيف ملف فرز دمج SORT MERGE FILE DESCRIPTION ENTRY

وهي جملـة ترد في فصل الملفات FILE SECTION من قسم البيانات تبدأ بمؤشر هو SD و يتبعها اسم الملف و يليها عبارات توصيف الملف .

برنامج مكتوب بصيغة المصدر SOURCE PROGRAM

وهو البرنامج المكتوب بلغة المصدر ويمكن أن يحتوي على رموز وأشكال خاصة ، لكنها تتبع قواعد لغة كوبول .

رموز خاصة SPECIAL CHARACTER

وهي الرموز التالية :

| الرمز | المعنى |
|-------|----------------------------------|
| + | إشارة الاضافة |
| - | إشارة الطرح |
| * | نجمة (إشارة ضرب) |
| / | إشارة الخط المائل (إشارة القسمة) |
| = | إشارة المساواة |
| \$ | إشارة العملة (الدولار) |
| , | الفاصلة |
| ; | الفاصلة المنقوطة |
| . | النقطة (الفاصلة العشرية) |
| » | علامة الاقتباس |
| (| قوس أيسر |
|) | قوس أيمن |

> إشارة أكبر من

< إشارة أصغر من

أسماء خاصة SPECIAL NAMES

وهي اسم فقرة في قسم المحيطات ENVIRONMENT DIVISION والتي تشير إلى أسماء معطاة من قبل المبرمج بدلاً من الأسماء الرمزية (المنطقية) المعروفة سابقاً .

تعليلة STATEMENT

مجموعة من الكلمات والاشارات المكتوبة بصيغة قاعدية صحيحة في قسم الاجراءات وتبدأ بفعل .

دليل SUBSCRIPT

وهو رقم صحيح تدل قيمته على موقع عنصر محدد في الجدول .

جدول TABLE

وهو مجموعة متتالية من عناصر البيانات تعرف في قسم البيانات بعبارة OCCURS

عنصر الجدول TABLE ELEMENT

وهو عنصر تابع لمجموعة عناصر مكررة تشكل جدول .

قيمة حقيقية TRUTH VALUE

وهي تمثيل لنتيجة تقويم شرط لقيمتين هما :

صحيح TRUE و FALSE خطأ

عامل الوحدة UNARY OPERATOR

وهي إشارة الموجب (+) أو إشارة السالب (-) التي تسبق متغير أو القوس
الأسري لتعير رياضي والتي لها تأثير الضرب بـ ١ + أو ١ - .

المتغير VARIABLE

وهو حقل يمكن لقيمته أن تتغير أثناء تنفيذ البرنامج ، والمتغير المستخدم في التعبير
الرياضي يجب أن يكون حقلاً رقمياً .

فعل VERB

وهي كلمة قد تدل على عمل يجب أن يتخذ بواسطة مترجم برنامج كوبول
COBOL COMPILER أو بواسطة برنامج مترجم إلى لغة الآلة .

الكلمة (في لغة كوبول) WORD

سلسلة من الرموز لا تزيد عن ٣٠ رمزاً وتشكل كلمة يستخدمها المبرمج في برنامج
كوبول أو اسم نظام أو كلمة محجوزة .

فصل التخزين الداخلي للبيانات WORKING-STORAGE SECTION

وهو فصل من قسم البيانات DATA DIVISION والذي يوصف عناصر البيانات
المستخدمة كمنطقة عمل في البرنامج ، هذه العناصر يمكن أن تكون عناصر مستقلة أو
عناصر بيانات مترابطة ومتفرعة أو يمكن أن تكون الاثنين معاً .

١٥-٥ حلول مقترحة لبعض التمرينات

* حل تمرين رقم ١ في الفصل ٣

IDENTIFICATION DIVISION.
PROGRAM-ID. TXT16.
AUTHOR. KHAYAT-JNAID.

*
* يقوم هذا البرنامج بقراءة ملف
* بطاقات مثقبة ويكتبه على القرص
* الممغنط.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT CARD-F ASSIGN TO UT-S-CARDF.
SELECT CONSMFL ASSIGN TO UT-S-CONF.

DATA DIVISION.
FILE SECTION.
FD CARD-F LABEL RECORD IS OMITTED.
01 CARD-REC PIC X(80).

*
FD CONSMFL LABEL RECORD IS STANDARD.
01 CON-REC PIC X(80).

PROCEDURE DIVISION.
OPEN INPUT CARD-F
OUTPUT CONSMFL.

* قراءة سجل من الملف
READ-CARD.
READ CARD-F AT END
CLOSE CARD-F CONSMFL STOP RUN.

* كتابه سجل على ملف الإخراج
MOVE CARD-REC TO CON-REC.
WRITE CON-REC.
GO TO READ-CARD.

```

*      حل تمرين رقم ٢ في الفصل ٣
IDENTIFICATION DIVISION.
PROGRAM-ID. TXT07.
AUTHOR.      KHAYAT-JNAID.

*
*      يقوم هذا البرنامج بكتابة ٧ سجل
*      على ملف اخراج اول ثم ٣ سجل اخر على
*      ملف اخراج ثاني باستخدام TIMES
*
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.  IBM-370.
OBJECT-COMPUTER.  IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
        SELECT INFILE  ASSIGN TO UT-S-INFILE.
        SELECT DFILE   ASSIGN TO UT-S-DFILE.
        SELECT TFILE   ASSIGN TO UT-S-TFILE.

DATA DIVISION.
FILE SECTION.
FD      INFILE          LABEL RECORD IS OMITTED
                        DATA RECORD IS IN-REC.
01      IN-REC          PIC          X(80).

*
FD      DFILE           LABEL RECORD IS STANDARD
                        DATA RECORD IS D-REC.
01      D-REC           PIC          X(80).

*
FD      TFILE           LABEL RECORD IS STANDARD
                        DATA RECORD IS T-REC.
01      T-REC           PIC          X(80).

PROCEDURE DIVISION.
        OPEN  INPUT  INFILE
              OUTPUT DFILE,  TFILE.

*
        PERFORM PAR-1 70 TIMES.
        PERFORM PAR-2 30 TIMES.
        CLOSE INFILE TFILE DFILE STOP RUN.

*
PAR-1.
        READ INFILE.

```

```

WRITE D-REC FROM IN-REC.

*
PAR-2.
  READ INFILE AT END GO TO EXT2.
  WRITE T-REC FROM IN-REC.
EXT2.
  EXIT.

*
*      حل تمرين رقم ٣ في الفصل ٥-٣
IDENTIFICATION DIVISION.
PROGRAM-ID. TXT05.
AUTHOR.      KHAYAT-JNAID.

*
*      يقوم هذا التمرين بحساب سعر السلعة
*      ويطبّع تقريراً بذلك
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
  SOURCE-COMPUTER.  IBM-370.
  OBJECT-COMPUTER.  IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
  SELECT C-FILE ASSIGN TO UT-S-CFILE.
  SELECT P-FILE ASSIGN TO UT-S-PFILE.

DATA DIVISION.
FILE SECTION.
FD      C-FILE      LABEL RECORD IS OMITTED
                     BLOCK CONTAINS 0 RECORDS
                     DATA RECORD IS C-REC.

01      C-REC.
        03      QTY          PIC 9(4)V99.
        03      PRICE        PIC 9(3)V99.
        03      FILLER        PIC X(69).

*
FD      P-FILE      LABEL RECORD IS OMITTED
                     DATA RECORD IS P-REC.

01      P-REC          PIC X(132).

*
WORKING-STORAGE SECTION.
01      HDR-1.

```

```

03 FILLER PIC X(20) VALUE SPACES.
03 FILLER PIC X(11) VALUE 'LIST VALUES'.
03 FILLER PIC X(101) VALUE SPACES.
01 HDR-2.
03 FILLER PIC X(03) VALUE SPACES.
03 FILLER PIC X(08) VALUE 'QUANTITY'.
03 FILLER PIC X(06) VALUE SPACES.
03 FILLER PIC X(11) VALUE 'PRICE'.
03 FILLER PIC X(04) VALUE SPACES.
03 FILLER PIC X(28) VALUE 'AMOUNT'.
01 DAATA-LINE.
03 FILLER PIC X(03) VALUE SPACES.
03 P-QNTY PIC 9(4).99.
03 FILLER PIC X(09) VALUE SPACES.
03 P-PRICE PIC 999.99.
03 FILLER PIC X(04) VALUE SPACES.
03 P-AMONT PIC Z9(8).99.
03 FILLER PIC X(91) VALUE SPACES.
01 TOTAL-LINE.
03 FILLER PIC X(29) VALUE 'TOTAL'.
03 P-TOTAL PIC ZZZZ9(5).99.
03 FILLER PIC X(91) VALUE SPACES.
* عددادات تستخدم للتجميع
01 TOTAL-AREAS.
03 AMONT-A PIC 9(9)V99 VALUE ZERO.
03 TOTAL-A PIC 9(9)V99 VALUE ZEROS.
03 LINE-CNTR PIC 99 VALUE 0.
PROCEDURE DIVISION.
OPEN INPUT C-FILE
OUTPUT P-FILE.
* طباعة العناوين
PRINT-HEADERS.
MOVE SPACES TO P-REC.
WRITE P-REC AFTER ADVANCING PAGE.
MOVE HDR-1 TO P-REC.
WRITE P-REC AFTER ADVANCING 3 LINES.
MOVE HDR-2 TO P-REC.
WRITE P-REC AFTER ADVANCING 2 LINES.
MOVE SPACES TO P-REC.
WRITE P-REC AFTER 1 LINE.
MOVE ZERO TO LINE-CNTR.

```

```

MAIN-LOOP.
  READ      C-FILE AT END
              GO TO TOTAL-P.
  MULTIPLY QTY BY PRICE GIVING AMONT-A.
  ADD      AMONT-A      TO      TOTAL-A.
  MOVE     AMONT-A      TO      P-AMONT.
  MOVE     QTY          TO      P-QNTY.
  MOVE     PRICE        TO      P-PRICE.
  MOVE     DAATA-LINE TO      P-REC.
  WRITE    P-REC        AFTER ADVANCING 1 LINE.
  ADD      1            TO      LINE-CNTR.
  IF       LINE-CNTR =    24
    GO TO   PRINT-HEADERS
  ELSE
    GO TO   MAIN-LOOP.
TOTAL-P.
  MOVE     TOTAL-A      TO      P-TOTAL.
  MOVE     TOTAL-LINE TO      P-REC.
  WRITE    P-REC        AFTER 3.
*
  CLOSE   C-FILE,      P-FILE.
  STOP    RUN.

```

| QUANTITY | LIST VALUES | |
|----------|-------------|-------------|
| | PRICE | AMOUNT |
| 0020.00 | 020.00 | 00000400.00 |
| 0034.00 | 012.00 | 00000408.00 |
| 0067.00 | 023.00 | 00001541.00 |
| 0050.00 | 034.00 | 00001700.00 |
| 0035.50 | 023.30 | 00000827.15 |
| 0025.50 | 045.50 | 00001160.25 |
| TOTAL | | 06036.40 |

شكل المخرجات

* حل تمرين رقم 1 في الفصل 11-1
 IDENTIFICATION DIVISION.
 PROGRAM-ID. TXT11.
 AUTHOR. KHAYAT-JNAID.
 *
 * يقوم هذا البرنامج بإنشاء ملف مستقل
 * باستخدام تعليمتي 'DISPLAY' & 'ACCEPT'
 *
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. IBM-370.
 OBJECT-COMPUTER. IBM-370.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
 SELECT PAYFILE ASSIGN TO UT-S-PPFILE.
 DATA DIVISION.
 FILE SECTION.
 FD PAYFILE LABEL RECORD IS STANDARD
 DATA RECORD IS PAY-REC.
 01 PAY-REC.
 03 EM-NO PIC 9(4).
 03 EM-NAME PIC A(20).
 03 EM-SAL PIC 9(5)V99.
 03 EM-TRANS PIC 9(3)V99.
 03 BL-NK PIC X(4).
 *
 PROCEDURE DIVISION.
 OPEN OUTPUT PAYFILE.
 *
 AC-NUMB.
 DISPLAY ' أدخل رقم الموظف:'.
 ACCEPT EM-NO.
 IF EM-NO = 9999
 CLOSE PAYFILE STOP RUN.
 IF EM-NO NOT NUMERIC OR
 EM-NO LESS THAN 1 GO TO AC-NUMB.
 AC-NAME.
 DISPLAY ' أدخل اسم الموظف:'.
 ACCEPT EM-NAME.
 IF EM-NAME = SPACE OR
 EM-NAME NOT ALPHABETIC
 GO TO AC-NAME.


```

AC-SAL.
  DISPLAY '      : الراتب الموظف:'.
  ACCEPT EM-SAL.
  IF EM-SAL NOT NUMERIC GO TO AC-SAL.
AC-TRANSP.
  DISPLAY '      : النقل بدل:'.
  ACCEPT EM-TRANS.
  IF EM-TRANS NOT NUMERIC
    GO TO AC-TRANSP.

```

*

```

  WRITE    PAY-REC.
  GO TO    AC-NUMB.

```

* حل تمرين رقم ٢ في الفصل ١١-٤

```

IDENTIFICATION DIVISION.
PROGRAM-ID. TXT10.
AUTHOR.      KHAYAT-JNAID.

```

*

* يقوم هذا البرنامج بحساب صافي الراتب

*

وطباعته.

```

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
  SOURCE-COMPUTER.  IBM-370.
  OBJECT-COMPUTER.  IBM-370.
  SPECIAL-NAMES.    DECIMAL-POINT IS COMMA.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
  SELECT PAYFILE ASSIGN TO UT-S-PFILE.
  SELECT P-FILE ASSIGN TO UT-S-PRFL.
DATA DIVISION.
FILE SECTION.
FD    PAYFILE    LABEL RECORD IS STANDARD
      BLOCK CONTAINS 0 RECORDS
      DATA RECORD IS PAY-REC.

```

```

01    PAY-REC.
      03    EM-NO      PIC 9(4).
      03    EM-NAME    PIC X(20).
      03    EM-SAL     PIC 9(5)V99.
      03    EM-TRANS   PIC 9(3)V99.
      03    FILLER     PIC X(44).

```

```

*
FD      P-FILE      LABEL RECORD IS OMITTED
                        DATA RECORD IS P-REC.
01      P-REC        PIC      X(132).
*
WORKING-STORAGE SECTION.
01      HDR-1.
03      FILLER       PIC      X(56)    VALUE SPACES.
03      FILLER       PIC      X(19)    VALUE
                                'BRIGHT STAR COMPANY'.
03      FILLER       PIC      X(57)    VALUE SPACES.
01      HDR-2.
03      FILLER       PIC      X(10)    VALUE
                                ' PAGE NO :'.
03      P-CTR        PIC      ZZ9.
03      FILLER       PIC      X(44)    VALUE SPACES.
03      FILLER       PIC      X(15)    VALUE
                                ' PAYROLL REPORT'.
03      FILLER       PIC      X(44)    VALUE SPACES.
03      FILLER       PIC      X(06)    VALUE 'DATE:'.
03      DAY-P        PIC      99.
03      FILLER       PIC      X      VALUE '/'.
03      MONTH-P      PIC      99.
03      FILLER       PIC      XXX     VALUE '/19'.
03      YEAR-P       PIC      99.
03      FILLER       PIC      X(3)    VALUE SPACES.
01      HDR-3.
03      FILLER       PIC      X(20)
                                VALUE ' EMPLOYEE NO.'.
03      FILLER       PIC      X(20)
                                VALUE 'EMPLOYEE NAME'.
03      FILLER       PIC      X(14)
                                VALUE 'BASE SALARY'.
03      FILLER       PIC      X(16)
                                VALUE 'TRANSPORTATION'.
03      FILLER       PIC      X(15)
                                VALUE 'SOCIAL SECURITY'.
03      FILLER       PIC      X(13)
                                VALUE ' SOCIAL-FUND'.
03      FILLER       PIC      X(13)
                                VALUE ' NET PAY'.

```

```

03 FILLER PIC X(22) VALUE ' SIGNATURE'.
01 DATA-LINE.
03 FILLER PIC X(06) VALUE SPACES.
03 P-NO PIC ZZ9.
03 FILLER PIC X(06) VALUE SPACES.
03 P-NAME PIC X(20).
03 FILLER PIC X(04) VALUE SPACES.
03 P-SAL PIC Z(4)9,99.
03 FILLER PIC X(08) VALUE SPACES.
03 P-TRANS PIC ZZ9,99.
03 FILLER PIC X(09) VALUE SPACES.
03 P-SOCIAL PIC Z(3)9,99.
03 FILLER PIC X(10) VALUE SPACES.
03 P-FUNDS PIC Z(3)9,99.
03 FILLER PIC X(07) VALUE SPACES.
03 P-NET PIC Z(5)9,99.
03 FILLER PIC X(22) VALUE SPACES.
01 TOTAL-LINE.
03 FILLER PIC X(29) VALUE ' TOTAL'.
03 P-TOTAL PIC 9(4).99.
03 FILLER PIC X(91) VALUE SPACES.
* قبل التاريخ من الحساب
01 CURENT-DATE.
03 C-YEAR PIC 99.
03 C-MONTH PIC 99.
03 C-DAY PIC 99.
* أعداد تستخدم للتجميع
77 SOCIAL PIC 9(4)V99.
77 FUNDS PIC 9(4)V99.
77 NETPAY PIC 9(6)V99.
77 L-CON PIC 99 VALUE 0.
77 P-CNTR PIC 999 VALUE 0.
PROCEDURE DIVISION.
OPEN INPUT PAYFILE
OUTPUT P-FILE.
* ACCEPT DATE & PRINT HEADERS.
ACCEPT CURENT-DATE FROM DATE.
MOVE C-YEAR TO YEAR-P.
MOVE C-MONTH TO MONTH-P.
MOVE C-DAY TO DAY-P.
PERFORM NEW-PAGE.

```

```

*
MAIN-LOOP.
    READ      PAYFILE      AT END
    CLOSE PAYFILE P-FILE STOP RUN.

*
*   حساب التامينات الاجتماعية :
MULTIPLY EM-SAL BY 0,05 GIVING SOCIAL.

*
*
*   معالجه حاله المرتب اقل من 1000,01 ريال :
IF      EM-SAL      (      1000,01
MULTIPLY EM-SAL BY 0,01 GIVING FUNDS
ELSE

*
*
*   معالجه حاله المرتب بين 1000 و 2000 ريال :
IF EM-SAL > 1000,01 AND EM-SAL < 2000,01
    COMPUTE FUNDS =
        ( EM-SAL - 1000 ) * 0,03 + 10
ELSE

*
*   معالجه حاله المرتب اكبر من 2000 ريال :
*   ملاحظه : تعتبر 10 و 30 هم حاصل النسبه عن
*   الاولين والاولين.
COMPUTE FUNDS =
    ( EM-SAL - 2000 ) * 0,05 + 10 + 30.

*
*
*   حساب صافي المرتب :
COMPUTE NETPAY =
    EM-SAL + EM-TRANS - ( FUNDS + SOCIAL ).

*
*
*   تمرير الحقول والنتائج الى سطر الطباعة :
MOVE      EM-NO      TO      P-NO.
MOVE      EM-NAME     TO      P-NAME.

```

```

MOVE    EM-SAL    TO    P-SAL.
MOVE    EM-TRANS  TO    P-TRANS.
MOVE    SOCIAL    TO    P-SOCIAL.
MOVE    FUNDS     TO    P-FUNDS.
MOVE    NETPAY    TO    P-NET.

```

*

*

* : كتابه السجل على ملف الطباعة

```

WRITE P-REC FROM DATA-LINE
      AFTER ADVANCING 2 LINES.
ADD 1 TO L-CON.
IF L-CON > 25
    PERFORM NEW-PAGE.
GO TO MAIN-LOOP.

```

*

*

* ذكره قلب الصفحة وكتابته العناوين:

```

NEW-PAGE.
MOVE ZERO TO L-CON.
ADD 1 TO P-CNTR.
MOVE P-CNTR TO P-CTR.
MOVE SPACES TO P-REC.
WRITE P-REC AFTER PAGE.
WRITE P-REC FROM HDR-1 AFTER 1 LINE.
WRITE P-REC FROM HDR-2 AFTER 1 LINE.
WRITE P-REC FROM HDR-3 AFTER 2 LINE.
EXIT.
EXIT.

```

أنظر شكل المخرجات صفحة ٥٩٩

* حل تمرين رقم ٣ في الفصل ١١-٤
 IDENTIFICATION DIVISION.
 PROGRAM-ID. TXT08.
 AUTHOR. KHAYAT-JNAID.
 *
 * يقوم هذا البرنامج بطباعة درجه وتقدير
 * الطالب مستخدم
 *
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. IBM-370.
 OBJECT-COMPUTER. IBM-370.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
 SELECT STDFILE ASSIGN TO UT-S-STDf.
 SELECT PRINTF ASSIGN TO UT-S-PRFL.
 DATA DIVISION.
 FILE SECTION.
 FD STDFILE LABEL RECORD IS OMITTED
 BLOCK CONTAINS 0 RECORDS.
 01 STD-REC.
 03 STD-NAME PIC X(25).
 03 STD-MARK PIC 999.
 88 FAIL VALUES ARE 00 THRU 59.
 88 PASS VALUES ARE 60 THRU 69.
 88 GOOD VALUES ARE 70 THRU 79.
 88 VRYGOOD VALUES ARE 80 THRU 89.
 88 EXCELNT VALUES ARE 90 THRU 100.
 03 FILLER PIC X(52).
 *
 FD PRINTF LABEL RECORD IS OMITTED.
 01 P-REC.
 03 FILLER PIC X(3).
 03 S-NAME PIC X(25).
 03 FILLER PIC X(6).
 03 S-MARK PIC 9(3).
 03 FILLER PIC X(4).
 03 S-GRAD PIC X(7).
 03 FILLER PIC X(84).
 WORKING-STORAGE SECTION.
 77 END-OF-JOB PIC XXX VALUE 'NO'.

```

PROCEDURE DIVISION.
    OPEN      INPUT   STDFILE
              OUTPUT  PRINTF.
* قراءة سجل من الملف
    READ      STDFILE  AT   END
              MOVE 'YES' TO END-OF-JOB.
    PERFORM WRITE-READ THRU EXT
              UNTIL END-OF-JOB = 'YES'.
    CLOSE STDFILE, PRINTF.
    STOP RUN.
* كتابة سجل على ملف الإخراج و قراءه سجل جديد
    WRITE-READ.
    IF FAIL      MOVE      'FAIL'      TO S-GRAD.
    IF PASS      MOVE      'PASS'      TO S-GRAD.
    IF GOOD      MOVE      'GOOD'      TO S-GRAD.
    IF VRYGOOD   MOVE      'V.GOOD'    TO S-GRAD.
    IF EXCELNT   MOVE      'EXCELL.'   TO S-GRAD.
    MOVE STD-NAME TO S-NAME.
    MOVE STD-MARK TO S-MARK.
    WRITE P-REC  AFTER 2.
    READ STDFILE AT   END
              MOVE      'YES' TO END-OF-JOB.
EXT.
EXIT.

```

شكل المخرجات

| | | |
|--------------------|-----|---------|
| ZUHAIR KHAYAT | 099 | EXCELL. |
| SAMI ALHALABI | 090 | EXCELL. |
| MOJAHED ALHARBI | 050 | FAIL |
| BADR ALJUNDI | 055 | FAIL |
| SUBHI RAJAB | 073 | GOOD |
| AHMAD HAMEED GHAZI | 078 | GOOD |
| FERAS AHMED JNAID | 084 | V.GOOD |

| | | |
|--------------------|-----|---------|
| KHALDOON AHMED | 085 | V.GOOD |
| SAMEER HUSNI | 070 | GOOD |
| MANSUR AL GHADAN | 060 | PASS |
| ABDULAZIZ QUAIZ | 063 | PASS |
| SALEM KHANAWATI | 065 | PASS |
| BASHIR ZUNDAH | 069 | PASS |
| ALI AL YUSEF | 057 | FAIL |
| MOHAMED ZAKI | 053 | FAIL |
| MOLHAM ZAINI | 089 | V.GOOD |
| AHMED JAMALI | 099 | EXCELL. |
| ABDULRAHMAN HARBI | 098 | EXCELL. |
| ABDUL AZIZ MANSOOR | 064 | PASS |
| GHIAS ALCHATTA | 065 | PASS |
| HUSNI AL SHMEMRI | 087 | V.GOOD |
| MONEER QUATARI | 085 | V.GOOD |
| NASER AL OHIBI | 084 | V.GOOD |
| OTHMAN AL BASHEER | 080 | V.GOOD |
| ALMAJED YAHYAWI | 083 | V.GOOD |
| MAAMAR MAKID | 080 | V.GOOD |
| LINAH KHAYAT | 095 | EXCELL. |
| YOUSUF QABOLI | 014 | FAIL |

* حل تمرين رقم ١ في الفصل ٥-١ *

IDENTIFICATION DIVISION.

PROGRAM-ID. TXT09.

AUTHOR. KHAYAT-JNAID.

■

* يقوم هذا البرنامج بعمل تحقق على

* ملف الإدخال ويطبغ قائمه بالاطفاء

*

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER. IBM-370.

OBJECT-COMPUTER. IBM-370.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT SUBFILE ASSIGN TO UT-S-SFIL.

SELECT P-FILE ASSIGN TO UT-S-PFIL.

SELECT INVFILE ASSIGN TO UT-S-IFIL.

SELECT VALFILE ASSIGN TO UT-S-VFIL.

DATA DIVISION.

FILE SECTION.

*

* توصيف ملف الإدخال :

FD SUBFILE LABEL RECORD IS STANDARD

BLOCK CONTAINS 0 RECORDS

DATA RECORD IS SUB-REC.

01 SUB-REC.

03 D-NO PIC 9(5).

03 D-NAME PIC X(20).

03 D-ADDRESS PIC X(24).

03 D-DATE.

05 D-DAY PIC 9(2).

88 V-DAY VALUES ARE 01 THRU 30.

05 D-MNTH PIC 9(2).

88 V-MNTH VALUES ARE 01 THRU 12.

05 D-YEAR PIC 9(4).

88 V-YEAR VALUES ARE 1402 THRU 1410.

03 D-CODE PIC 99.

88 V-CODE VALUES ARE 01 THRU 10.

03 D-AREA PIC 99.

88 V-AREA VALUES ARE 01 THRU 15.

03 D-CNTR PIC 9(7).

```

03 D-LREAD      PIC 9(7).
03 FILLER       PIC X(5).
*
* : توصيف ملف الطباعة
FD P-FILE LABEL RECORD IS OMITTED
DATA RECORD IS P-REC.
01 P-REC        PIC X(132).
*
*
* : توصيف ملف السجلات الخطا
FD INVFILE LABEL RECORD IS STANDARD
DATA RECORD IS INV-REC.
01 INV-REC      PIC X(80).
*
*
* : توصيف ملف السجلات الصحيحة
FD VALFILE LABEL RECORD STANDARD
DATA RECORD IS VAL-REC.
01 VAL-REC      PIC X(80).
*
WORKING-STORAGE SECTION.
*
01 HDR-1.
03 FILLER PIC X(58) VALUE SPACES.
03 FILLER PIC X(16) VALUE
'ELECTRIC COMPANY'.
03 FILLER PIC X(58) VALUE SPACES.
*
01 HDR-2.
03 FILLER PIC X(12) VALUE
'DATE: '.
03 DAY-P PIC 99.
03 FILLER PIC X VALUE '/'.
03 MNTH-P PIC 99.
03 FILLER PIC XXX VALUE '/19'.
03 YEAR-P PIC 99.
03 FILLER PIC X(29) VALUE SPACES.
03 FILLER PIC X(34) VALUE
'LIST OF INVALID SUBSCRIBER RECORDS'.
03 FILLER PIC X(34) VALUE SPACES.
03 FILLER PIC X(99) VALUE
'PAGE NO : '.

```

| | | | | |
|----|------------------------|-----|---------|---------------|
| 03 | P-CTR | PIC | ZZ9. | |
| * | | | | |
| 01 | HDR-3. | | | |
| 03 | FILLER | PIC | X(20) | VALUE |
| | ' SUBSCRIBER NO.' | | | |
| 03 | FILLER | PIC | X(20) | VALUE |
| | 'SUBS. N A M E ' | | | |
| 03 | FILLER | PIC | X(27) | VALUE |
| | 'SUBS. A D D R E S S ' | | | |
| 03 | FILLER | PIC | X(20) | VALUE |
| | 'SUBSCRIPTION DATE' | | | |
| 03 | FILLER | PIC | X(13) | VALUE |
| | ' SUB. C O D E ' | | | |
| 03 | FILLER | PIC | X(12) | VALUE |
| | ' SUB. AREA ' | | | |
| 03 | FILLER | PIC | X(20) | VALUE |
| | ' LAST READING ' | | | |
| * | | | | |
| 01 | DETAL-L. | | | |
| 03 | FILLER | PIC | X(04) | VALUE SPACES. |
| 03 | D-NO | PIC | XXXXX. | |
| 03 | FILLER | PIC | X(07) | VALUE SPACES. |
| 03 | D-NAME | PIC | X(20). | |
| 03 | FILLER | PIC | X(03) | VALUE SPACES. |
| 03 | D-ADRESS | PIC | X(24). | |
| 03 | FILLER | PIC | X(10) | VALUE SPACES. |
| 03 | D-DATE. | | | |
| | 05 D-YEAR | PIC | 9(4). | |
| | 05 FILLER | PIC | X | VALUE '/'. |
| | 05 D-MNTH | PIC | 9(2). | |
| | 05 FILLER | PIC | X | VALUE '/'. |
| | 05 D-DAY | PIC | 9(2). | |
| 03 | FILLER | PIC | X(05) | VALUE SPACES. |
| 03 | D-CODE | PIC | 99. | |
| 03 | FILLER | PIC | X(13) | VALUE SPACES. |
| 03 | D-AREA | PIC | 99. | |
| 03 | FILLER | PIC | X(09) | VALUE SPACES. |
| 03 | D-CNTR | PIC | Z(5)99. | |
| 03 | FILLER | PIC | X | VALUE SPACES. |
| 03 | D-LREAD | PIC | Z(5)99. | |
| 03 | FILLER | PIC | X | VALUE SPACES. |

```

*
01    AST-L.
03    FILLER    PIC    X(06)    VALUE SPACES.
03    AST-1    PIC    X.
03    FILLER    PIC    X(18)    VALUE SPACES.
03    AST-2    PIC    X.
03    FILLER    PIC    X(24)    VALUE SPACES.
03    AST-3    PIC    X.
03    FILLER    PIC    X(22)    VALUE SPACES.
03    AST-4    PIC    X.
03    FILLER    PIC    X(04)    VALUE SPACES.
03    AST-5    PIC    X.
03    FILLER    PIC    X(03)    VALUE SPACES.
03    AST-6    PIC    X.
03    FILLER    PIC    X(07)    VALUE SPACES.
03    AST-7    PIC    X.
03    FILLER    PIC    X(12)    VALUE SPACES.
03    AST-8    PIC    X.
03    FILLER    PIC    X(09)    VALUE SPACES.
03    AST-9    PIC    X.
03    FILLER    PIC    X(08)    VALUE SPACES.
03    AST-10   PIC    X.
03    FILLER    PIC    X(03)    VALUE SPACES.

```

```

*
*    لقبول التاريخ من الماسك:

```

```

01    CURENT-DATE.
03    C-YEAR    PIC    99.
03    C-MNTH    PIC    99.
03    C-DAY     PIC    99.

```

```

*
*    الحقل التالي يستخدم لاختبار نهاية الملف:
77    EOFL      PIC    XXX      VALUE SPACES.

```

```

*
*    عدادات مستخدم للتمديد:
77    LINES-CONT PIC    99      VALUE 25.
77    FLAG       PIC    9       VALUE ZERO.
77    PAGE-CNTR  PIC    99      VALUE ZERO.
77    LINE-NO    PIC    9999    VALUE 0.

```

```

*
PROCEDURE DIVISION.
    OPEN      INPUT      SUBFILE
            OUTPUT      P-FILE, INVFILE VALFILE.

```

```

*
*      قبول التاريخ وطباعة العناوين
*      وقراءه السجل الاول من الملف
*
      ACCEPT      CURENT-DATE FROM DATE.
      MOVE        C-YEAR      TO   YEAR-P.
      MOVE        C-MNTH      TO   MNTH-P.
      MOVE        C-DAY       TO   DAY-P.
      PERFORM     NEW-PAGE     THRU EXTP.
      READ        SUBFILE      AT END
      MOVE        'END' TO EOFL.

*
*      المعالجه الرئيسيه للبرنامج
      PERFORM MAIN-LOOP THRU END-LOOP
      UNTIL        EOFL = 'END'.
      PERFORM     END-RTN     THRU EXT-END.
      STOP        RUN.

*
*
*      عمل تدقيق على كل سجل وتحويله نجمه
*      تمت التدقيق غير الصحيح :
*
      MAIN-LOOP.
      MOVE        ZERO        TO FLAG.
      MOVE        SPACES      TO AST-L.
      IF D-NO      OF SUB-REC NOT NUMERIC OR
      D-NO        OF SUB-REC < 1
      MOVE 'X' TO AST-1.

*
      IF D-NAME    OF SUB-REC = SPACE OR
      D-NAME      OF SUB-REC NUMERIC
      MOVE 'X' TO AST-2.

*
      IF D-ADRESS  OF SUB-REC = SPACE
      MOVE 'X' TO AST-3.

*
      IF          NOT V-YEAR OF SUB-REC
      MOVE 'X' TO AST-4.

*
      IF          NOT V-MNTH OF SUB-REC
      MOVE 'X' TO AST-5.

```

```

*      IF      NOT V-DAY   OF SUB-REC
                                         MOVE '*' TO AST-6.
*
*      IF      NOT V-CODE  OF SUB-REC
                                         MOVE '*' TO AST-7.
*
*      IF NOT V-AREA  OF SUB-REC
                                         MOVE '*' TO AST-8.
*
*      IF D-CNTR      OF SUB-REC NOT NUMERIC
OR D-CNTR      OF SUB-REC LESS  THAN 1
                                         MOVE '*' TO AST-9.
*
*      IF D-LREAD     OF SUB-REC NOT NUMERIC
                                         MOVE '*' TO AST-10.
*
      IF AST-L = SPACES
        WRITE VAL-REC      FROM SUB-REC
                                MOVE 1 TO FLAG.
      IF FLAG = 1          NEXT SENTENCE
      ELSE
      IF LINES-CONT > 24
        PERFORM NEW-PAGE THRU EXTP
      ELSE
        ADD 1              TO LINES-CONT
        MOVE CORRESPONDING SUB-REC
                                TO DETAL-L
        WRITE P-REC        FROM DETAL-L AFTER 1
        WRITE P-REC        FROM AST-L  AFTER 1
        WRITE INV-REC FROM SUB-REC.
      READ SUBFILE         AT END
                                MOVE 'END ' TO EOFL.
END-LOOP.
EXIT.
*
*      : فخره قلب المرفعه وطبع العناوين :
NEW-PAGE.
      MOVE SPACES          TO P-REC.
      WRITE P-REC          AFTER PAGE.
      MOVE ZERO            TO LINES-CONT.
      ADD 1                TO PAGE-CNTR.

```

```

        MOVE PAGE-CNTR TO      P-CTR.
        WRITE P-REC FROM HDR-1 AFTER 1 LINES.
        WRITE P-REC FROM HDR-2 AFTER 1 LINES.
        WRITE P-REC FROM HDR-3 AFTER 1 LINES.
EXTP.
        EXIT.
*
*
*   فقره نهایی:
END-RTN.
        CLOSE SUBFILE VALFILE, INVFILE, P-FILE.
EXT-END.
        EXIT.

```

أنظر شكل المخرجات صفحة ٦٠٠

```

*   حل تمرين رقم ٢ في الفصل ٥-١
IDENTIFICATION DIVISION.
PROGRAM-ID. TXT12.
AUTHOR.      KHAYAT-JNAID.
*
*   يقوم هذا البرنامج باستدعاء برنامج
*   باستدعاء برنامج ذرعي آخر وذلك لاستدعاء
*   قسط البرنامج
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.  IBM-370.
OBJECT-COMPUTER.  IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
        SELECT CARD-F ASSIGN TO UT-S-CARD.
        SELECT PRINTF ASSIGN TO UT-S-PRFL.
DATA DIVISION.
FILE SECTION.
FD      CARD-F          LABEL RECORD IS OMITTED
                        DATA RECORD IS CARD-REC.
01      CARD-REC.
        03      EM-NO      PIC 9(5).
        03      EM-NAME    PIC X(30).

```

```

03      EM-SAL      PIC 9(5)V99.
03      EM-TRANS    PIC 9(3)V99.
03      EM-TRAIN     PIC 9(5)V99.
03      EM-DATE.
05      EM-DAY       PIC 99.
05      EM-MNTH      PIC 99.
05      EM-YEAR      PIC 9999.
03      FILLER       PIC X(18).

*
FD      PRINTF      LABEL RECORD IS OMITTED
                        DATA RECORD IS P-REC.
01      P-REC        PIC X(132).

WORKING-STORAGE SECTION.
*   الدخول التاليه ستذهب الى البرنامج
*   الفرعي بتعليقه CALL ليتم الحساب بها
01      CALC.
03      W-SAL        PIC 9(5)V99.
03      W-TRANS      PIC 9(3)V99.
03      W-TRAIN      PIC 9(5)V99.
03      W-NET        PIC 9(5)V99.

*
01      D-L.
03      FILLER PIC X(05) VALUE SPACES.
03      P-NO PIC 9(5).
03      FILLER PIC X(05) VALUE SPACES.
03      P-NAME PIC X(30).
03      FILLER PIC X(05) VALUE SPACES.
03      P-SAL PIC Z(4)9.99.
03      FILLER PIC X(05) VALUE SPACES.
03      P-TRAN PIC ZZ9.99.
03      FILLER PIC X(09) VALUE SPACES.
03      P-TRIN PIC Z(4)9.99.
03      FILLER PIC X(05) VALUE SPACES.
03      P-NET PIC Z(4)9.99.
03      FILLER PIC X(05) VALUE SPACES.

PROCEDURE DIVISION.
OPEN      INPUT      CARD-F
                        OUTPUT PRINTF.

*
MAIN-PROG.
READ      CARD-F      AT END
CLOSE CARD-F PRINTF STOP RUN.

```



```

*
*   ارسال السجل المذكور مع مساحات العمل CALC
*   الى البرنامج الفرعي SUBPGM للحساب :
*
*   CALL 'TXTSUB' USING CARD-REC CALC.
*
*   بعد تنفيذ التعليمة اعلاه تاتي المدخل
*   لتحرك الى سطر الطباعة
*
*   MOVE      EM-NO      TO      P-NO.
*   MOVE      EM-NAME    TO      P-NAME
*   MOVE      W-SAL      TO      P-SAL.
*   MOVE      W-TRANS    TO      P-TRAN.
*   MOVE      W-TRAIN    TO      P-TRIN.
*   MOVE      W-NET      TO      P-NET.
*
*   WRITE      P-REC      FROM      D-L AFTER 2.
*   GO TO      MAIN-PROG.

```

```

*   هل تمرين رقم 3 في الفصل ١٠-٥
*   تابع لمل البرنامج TXT12
*   IDENTIFICATION DIVISION.
*   PROGRAM-ID. TXTSUB.
*   AUTHOR.      KHAYAT-JNAID.
*
*   هذا البرنامج المستدعي من قبل البرنامج
*   TXT12 ويقوم باحتساب قسط الراتب
*
*   ENVIRONMENT DIVISION.
*   CONFIGURATION SECTION.
*   SOURCE-COMPUTER.  IBM-370.
*   OBJECT-COMPUTER.  IBM-370.
*   DATA DIVISION.
*   WORKING-STORAGE SECTION.
*   77  W-DAYS          PIC  99.
*   LINKAGE SECTION.
*   المدخل التاليه تستقبل مايقابلها من
*   مدخل مرث لهذا البرنامج بتعليمة CALL
*   عبر قسم LINKAGE SECTION
*   01  L-REC.
*       03  L-NO        PIC  9(5).
*       03  L-NAME      PIC  X(30).

```

```

03      L-SAL      PIC  9(5)V99.
03      L-TRANS    PIC  9(3)V99.
03      L-TRAIN    PIC  9(5)V99.
03      L-DATE.
05      L-DAY      PIC  99.
05      L-MNTH     PIC  99.
05      L-YEAR     PIC  9999.
03      FILLER     PIC  X(18).
01  WS-CALC.
03      WS-SAL      PIC  9(5)V99.
03      WS-TRANS    PIC  9(3)V99.
03      WS-TRAIN    PIC  9(5)V99.
03      WS-NET      PIC  9(5)V99.

```

*
 * : USING تغيير عنوان القسم التالي واستخدام
 PROCEDURE DIVISION USING L-REC WS-CALC.

```

*
SUBTRACT L-DAY FROM 30 GIVING W-DAYS.
COMPUTE WS-SAL      ROUNDED =
      L-SAL      / 30 * W-DAYS.
COMPUTE WS-TRANS    ROUNDED =
      L-TRANS    / 30 * W-DAYS.
COMPUTE WS-TRAIN    ROUNDED =
      L-TRAIN    / 30 * W-DAYS.
ADD      WS-SAL
      WS-TRANS
      WS-TRAIN
GIVING  WS-NET.
RETURN-TO-MAIN.
EXIT PROGRAM.

```

* العودة للبرنامج الرئيسي
 * TXT12

* حل تمرين رقم ١ في الفصل V-ع
 IDENTIFICATION DIVISION.
 PROGRAM-ID. TXT14.
 AUTHOR. KHAYAT-JNAID.

*
 * يقوم هذا البرنامج بدمج ثلاث ملفات
 * ادخال بملف اخراج واحد ثم تتم
 * معالجة الملف الناتج وطباعه محتواه

```

*
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
    SOURCE-COMPUTER.    IBM-370.
    OBJECT-COMPUTER.    IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT INF1      ASSIGN TO UT-S-INF1.
    SELECT INF2      ASSIGN TO UT-S-INF2.
    SELECT INF3      ASSIGN TO UT-S-INF3.
    SELECT MRGF      ASSIGN TO UT-S-MRGF.
    SELECT OTMRG     ASSIGN TO UT-S-OTMRG.
    SELECT PRINTF    ASSIGN TO UT-S-PRFL.

DATA DIVISION.
FILE SECTION.
*
FD      INF1 LABEL RECORD IS STANDARD
        RECORD CONTAINS 80 CHARACTERS
        DATA RECORD IS IN-REC1.
01      IN-REC1      PIC X(80).
*
FD      INF2 LABEL RECORD IS STANDARD
        RECORD CONTAINS 80 CHARACTERS
        DATA RECORD IS IN-REC2.
01      IN-REC2      PIC X(80).
*
FD      INF3 LABEL RECORD IS STANDARD
        RECORD CONTAINS 80 CHARACTERS
        DATA RECORD IS IN-REC3.
01      IN-REC3      PIC X(80).
*
SD      MRGF
        RECORD CONTAINS 80 CHARACTERS
        DATA RECORD IS MRG-REC.
01      MRG-REC.
03 MRG-NO      PIC 9(6).
03 MRG-DATA    PIC X(74).
*
FD      OTMRG LABEL RECORD STANDARD
        RECORD CONTAINS 80 CHARACTERS
        DATA RECORD IS OUT-REC.

```

```

01      OUT-REC.
03 CST-NO      PIC      9(6).
03 CST-NAME    PIC      X(24).
03 CST-BAL     PIC      S9(8)V99.
03 CST-FLRE    PIC      X(40).
*
FD      PRINTF      LABEL RECORD IS OMITTED
                        DATA RECORD IS P-REC.
01      P-REC      PIC      X(132).
*
WORKING-STORAGE SECTION.
01      HDR1.
03      FILLER    PIC      X(57)    VALUE SPACES.
03      FILLER    PIC      X(16)    VALUE
                        'CUSTOMMER REPORT'.
03      FILLER    PIC      X(57)    VALUE SPACES.
01      HDR2.
03      FILLER    PIC      X(08)    VALUE
                        ' DATE :'.
03      P-DATE    PIC      99/99/99.
03      FILLER    PIC      X(100)   VALUE SPACES.
03      FILLER    PIC      X(07)    VALUE
                        'PAGE :'.
03      P-PAGE    PIC      ZZ9.
03      FILLER    PIC      X(5)     VALUE SPACES.
01      HDR3.
03      FILLER    PIC      X(40)    VALUE
                        ' CUSTOMMER NO.'.
03      FILLER    PIC      X(21)    VALUE
                        ' CUSTOMMER NAME'.
03      FILLER    PIC      X(21)    VALUE
                        ' BALANCE'.
03      FILLER    PIC      X(50)    VALUE
                        ' COMMENTS'.
*
01      DATA-LINE.
03      FILLER    PIC      X(06)    VALUE SPACES.
03      P-NO      PIC      ZZZZ99.
03      FILLER    PIC      X(26)    VALUE SPACES.
03      P-NAME    PIC      X(24).
03      FILLER    PIC      X(15)    VALUE SPACES.

```

```

03 P-BAL PIC +++++999.99.
03 FILLER PIC X(27) VALUE SPACES.
03 P-COMNT PIC X(7).
03 FILLER PIC X(10) VALUE SPACES.

```

```

*
*          حفظ عمل لقبول التاريخ
*

```

```

01 C-DATE PIC 9(6).
01 PAG-CTR PIC 999 VALUE ZERO.
01 LIN-CTR PIC 99 VALUE 25.
77 FLAG PIC 9 VALUE ZERO.

```

```

*
PROCEDURE DIVISION.

```

```

*      تعليمه دمج الملفات الثلاث في ملف
*      استخراج واحد.
*      OPEN      تعليمه بفتح تعليمه
*

```

```

      MERGE MRGF
      ASCENDING KEY MRG-ND
      USING INF1 INF2 INF3
      GIVING OTMRG.

```

```

*      بعد تنفيذ تعليمه الدمج السابقة
*      تبدأ معالجة الملف OTMRG معالجه
*      عادية.
*

```

```

      START-PROCESSING.
      OPEN INPUT OTMRG
      OUTPUT PRINTF.
      ACCEPT C-DATE FROM DATE.
      MOVE C-DATE TO P-DATE.
      READ OTMRG AT END
      MOVE 1 TO FLAG.

```

```

*
MAIN-LOOP.
      PERFORM MOVE-WRITE
      THRU EXT-MOVE-WRITE
      UNTIL FLAG EQUAL 1.
      CLOSE OTMRG PRINTF.
      STOP RUN.

```

```

*
MOVE-WRITE.
    IF      CST-BAL      NEGATIVE
        MOVE '*****' TO      P-COMNT
    ELSE
        MOVE SPACES      TO      P-COMNT.
    MOVE    CST-NO        TO      P-NO.
    MOVE    CST-NAME      TO      P-NAME.
    MOVE    CST-BAL       TO      P-BAL.
    IF      LIN-CTR       >      24
        PERFORM NEW-PAGE.
    WRITE   P-REC FROM DATA-LINE.
    ADD     1             TO      LIN-CTR.
    READ    OTMRG AT END
        MOVE 1           TO      FLAG.
EXT-MOVE-WRITE.
    EXIT.
NEW-PAGE.
    MOVE    SPACES        TO      P-REC.
    WRITE   P-REC        AFTER    PAGE.
    MOVE    ZERO          TO      LIN-CTR.
    ADD     1             TO      PAG-CTR.
    MOVE    PAG-CTR       TO      P-PAGE.
    WRITE   P-REC        FROM      HDR1
        AFTER 1 LINE.
    WRITE   P-REC        FROM      HDR2
        AFTER 1 LINE.
    WRITE   P-REC        FROM      HDR3
        AFTER 2 LINE.
EXTP.
    EXIT.

```

```

*
*
*      حل تمرين رقم ٢ في الفصل ٧-ع
IDENTIFICATION DIVISION.
PROGRAM-ID. TXT13.
AUTHOR.      KHAYAT-JNAID.
*
*      يقوم هذا البرنامج بفرز ملف الإدخال
*      وكتابته الملف المضروب على ملف آخر
*      ثم معالجته.

```

```

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
    SOURCE-COMPUTER.    IBM-370.
    OBJECT-COMPUTER.    IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT STOCKF ASSIGN TO UT-S-INF1.
    SELECT SRTFIL ASSIGN TO UT-S-INF3.
    SELECT OUTSRT ASSIGN TO UT-S-MRGF.
    SELECT PRINTF ASSIGN TO UT-S-PRFL.

DATA DIVISION.
FILE SECTION.
*
FD      STOCKF LABEL RECORD IS STANDARD
        BLOCK CONTAINS 0 RECORDS
        DATA RECORD IS STK-REC.
01      STK-REC          PIC X(80).

* :FD بدل SD ب SD
SD      SRTFIL
        RECORD CONTAINS 80 CHARACTERS
        DATA RECORD IS SRT-REC.

* لاحظ انه لا ضروره لتوصيف كل المدخول بل
* يكتفى بمفاتيح الفرز
01      SRT-REC.
03      SRT-NO           PIC 9(5).
03      SRT-DATA         PIC X(75).

*
FD      OUTSRT LABEL RECORD STANDARD
        RECORD CONTAINS 80 CHARACTERS
        DATA RECORD IS OUT-REC.
01      OUT-REC.
03      OUT-NO           PIC 9(5).
03      OUT-DESC         PIC X(20).
03      OUT-QTY          PIC 9(5).
03      OUT-CODE         PIC 9(3).
03      OUT-PRC          PIC 9(3)V99.
03      OUT-MIN          PIC 9(5).
03      FILLER           PIC X(37).

```

```

*
FD      PRINTF      LABEL RECORD IS OMITTED
                        DATA RECORD IS P-REC.
01      P-REC      PIC      X(132).
*
W
WORKING-STORAGE SECTION.
01      HDR1.
03      FILLER PIC      X(56)  VALUE SPACES.
03      FILLER PIC      X(18)  VALUE
                        'S T O C K L I S T'.
03      FILLER PIC      X(56)  VALUE SPACES.
*
01      HDR2.
03      FILLER PIC      X(08)  VALUE
                        ' DATE : '.
03      P-DATE PIC      99/99/99.
03      FILLER PIC      X(100) VALUE SPACES.
03      FILLER PIC      X(07)  VALUE
                        'PAGE : '.
03      P-PAGE PIC      ZZ9.
03      FILLER PIC      X(5)   VALUE SPACES.
*
01      HDR3.
03      FILLER PIC      X(40)  VALUE
                        ' ITEM NO.      ITEM DESCRIPTION'.
03      FILLER PIC      X(29)  VALUE
                        ' QUANTITY      ITEM CODE'.
03      FILLER PIC      X(17)  VALUE
                        ' ITEM PRICE'.
03      FILLER PIC      X(46)  VALUE
                        ' MINIMUM LEVEL      COMMENTS'.
*
01      DATA-LINE.
03      FILLER PIC      X(06)  VALUE SPACES.
03      P-NO PIC      ZZZZ99.
03      FILLER PIC      X(06)  VALUE SPACES.
03      P-DESC PIC      X(20).
03      FILLER PIC      X(09)  VALUE SPACES.
03      P-QNTY PIC      ZZZZ9.
03      FILLER PIC      X(09)  VALUE SPACES.

```



```

03 P-CODE PIC Z99.
03 FILLER PIC X(10) VALUE SPACES.
03 P-PRCE PIC ZZ9.99.
03 FILLER PIC X(10) VALUE SPACES.
03 P-MINI PIC Z(4)9.
03 FILLER PIC X(17) VALUE SPACES.
03 P-COMNT PIC X(7).
03 FILLER PIC X(05) VALUE SPACES.

```

```

*
*      حفل عمل لقبول التاريخ

```

```

*
01 C-DATE PIC 9(6).
01 PAG-CTR PIC 999 VALUE ZERO.
01 LIN-CTR PIC 99 VALUE 25.
77 FLAG PIC 9 VALUE ZERO.

```

```

*
PROCEDURE DIVISION.

```

```

*      تعليمة فرز ملف الاندخال حسب رقم المنتج :
*      لاحظ ان الملفات لم تفتح بتعليمة OPEN.
*

```

```

      SORT      SRTFIL
                ASCENDING KEY SRT-NO
      USING     STOCKF
                GIVING      OUTSRT.
*      بعد تنفيذ تعليمة الفرز السابقة
*      تبدأ معالجة الملف OUTSRT معالجة
*      عادية.
*

```

```

      START-PROCESSING.
      OPEN INPUT OUTSRT
      OUTPUT PRINTF.
      ACCEPT C-DATE FROM DATE.
      MOVE C-DATE TO P-DATE.
      READ OUTSRT AT END
      MOVE 1 TO FLAG.

```

```

*
      MAIN-LOOP.
      PERFORM MOVE-WRITE
                THRU EXT-MOVE-WRITE
      UNTIL FLAG EQUAL 1.
      CLOSE OUTSRT PRINTF.
      STOP RUN.

```

*

```

MOVE-WRITE.
  IF      OUT-QTY  NOT  > OUT-MIN
  MOVE    'REORDER' TO  P-COMNT
  ELSE
  MOVE    SPACES    TO  P-COMNT.
  MOVE    OUT-NO     TO  P-NO.
  MOVE    OUT-DESC   TO  P-DESC.
  MOVE    OUT-QTY    TO  P-QNTY.
  MOVE    OUT-CODE   TO  P-CODE.
  MOVE    OUT-PRC    TO  P-PRCE.
  MOVE    OUT-MIN    TO  P-MINI.
  IF      LIN-CTR    > 24
  PERFORM NEW-PAGE.
  WRITE   P-REC      FROM DATA-LINE.
  ADD     1          TO  LIN-CTR.
  READ   OUTSRT     AT END
  MOVE    1          TO  FLAG.
EXT-MOVE-WRITE.
  EXIT.
NEW-PAGE.
  MOVE    SPACES     TO  P-REC.
  WRITE   P-REC      AFTER PAGE.
  MOVE    ZERO       TO  LIN-CTR.
  ADD     1          TO  PAG-CTR.
  MOVE    PAG-CTR    TO  P-PAGE.
  WRITE   P-REC      FROM  HDR1
  AFTER 1 LINE.
  WRITE   P-REC      FROM  HDR2
  AFTER 1 LINE.
  WRITE   P-REC      FROM  HDR3
  AFTER 2 LINE.
EXTP.
  EXIT.

```

*

أنظر شكل المخرجات صفحة ٦٠٠

IDENTIFICATION DIVISION. حل الترميز رقم ١
 PROGRAM-ID. TXT32.
 AUTHOR. KHAYAT-JNAID. الفصل ٦-أ

ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. T1990.
 OBJECT-COMPUTER. T1990.

*
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
 SELECT INFIL1 ASSIGN TO INPUT "F1".
 SELECT INFIL2 ASSIGN TO INPUT "F2".
 SELECT OUTMRG ASSIGN TO OUTPUT "MG".

*
 DATA DIVISION.
 FILE SECTION.
 FD INFIL1 LABEL RECORD STANDARD.
 01 INREC1.
 03 FRST-NO PIC 9(5).
 03 FRST-DATA PIC X(35).

*
 FD INFIL2 LABEL RECORD STANDARD.
 01 INREC2.
 03 SCND-NO PIC 9(5).
 03 SCND-DATA PIC X(35).

*
 FD OUTMRG LABEL RECORD STANDARD.
 01 MRDREC.
 03 MRG-NO PIC 9(5).
 03 MRG-DATA PIC X(35).

*
 PROCEDURE DIVISION.
 STARTT.
 OPEN INPUT INFIL1 INFIL2
 OUTPUT OUTMRG.

*
 * قراءه سجل من كل من ملفي الإدخال

*
 PERFORM READF1.
 PERFORM READF2.

*
 * تكرار المقارنة الأساسية في البرنامج إلى أن
 * تنتهي سجلات ملفي الإدخال

*
 PERFORM COMPAR UNTIL FRST-NO = SCND-NO
 AND FRST-NO = HIGH-VALUE.
 CLOSE INFIL1 INFIL2 OUTMRG.
 STOP RUN.

*
 * مقارنة رقمي السجلين ونقل الأصغر إلى ملف الإخراج
 * ثم كتابته على ملف الإخراج هذا، بعدها يتم التكرار
 * من الملف المنتول سجله = الأصغر =

```

* COMPAR.
* IF FRST-NO LESS THAN SCND-NO
  PERFORM MOVE-1 THRU EXIT1
ELSE
  PERFORM MOVE-2 THRU EXIT2.
* تحريك الحقول من الملف الاول للملف الاخر اذ
* وقراءه سجل جديد
*
MOVE-1.
WRITE MRGREC FROM INREC1.
*
READF1.
READ INFIL1 AT END MOVE HIGH-VALUE TO FRST-NO
EXIT1.
EXIT.
* تحريك الحقول من الملف الثاني للملف الاخر اذ
* وقراءه سجل جديد
*
MOVE-2.
WRITE MRGREC FROM INREC2.
*
READF2.
READ INFIL2 AT END MOVE HIGH-VALUE TO SCND-NO.
EXIT2.
EXIT.

```

```

* حل تمرين رقم ١ الفصل ٣-٥
IDENTIFICATION DIVISION.
PROGRAM-ID. TXT03.
AUTHOR. KHAYAT-JNAID.

```

```

*
* يبين هذا التمرين كيفية استخدام
* PERFORM ...TIMES تعليمه
*

```

```

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
  SELECT PRINTF ASSIGN TO UT-S-PRFL.

```

DATA DIVISION.
 FILE SECTION.
 FD PRINTF LABEL RECORD IS OMITTED
 DATA RECORD IS P-REC.
 01 P-REC PIC X(132).

*
 WORKING-STORAGE SECTION.
 77 P-CNTR PIC 99.
 PROCEDURE DIVISION.
 OPEN OUTPUT PRINTF.
 *
 PERFORM PAR-1 5 TIMES.
 CLOSE PRINTF.
 STOP RUN.

*
 PAR-1.
 MOVE ZERO TO P-CNTR.
 PERFORM PAR-2 10 TIMES.
 MOVE SPACES TO P-REC.
 WRITE P-REC.
 EXT1.
 EXIT.

*
 PAR-2.
 ADD 1 TO P-CNTR.
 MOVE P-CNTR TO P-REC.
 WRITE P-REC.
 EXT2.
 EXIT.

● حل التمرين رقم ٢ الفصل ٣ - ٥

IDENTIFICATION DIVISION.
 PROGRAM-ID. CUSTOM.
 AUTHOR. KHAYAT-JNAID.
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. TI990.
 OBJECT-COMPUTER. TI990.
 SPECIAL-NAMES. DECIMAL-POINT IS COMMA.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
 SELECT DEPFIL ASSIGN TO INPUT "DF".
 SELECT PRINTF ASSIGN TO PRINT "PF".

DATA DIVISION.

FILE SECTION.

FD DEPFIL LABEL RECORDS ARE STANDARD.

01 CUSTREC.

03 D-CODE PIC X.
03 D-ACNO PIC 9(5).
03 D-ACNAME PIC X(30).
03 D-ENDBAL PIC 9(7)V99.
03 D-DFST PIC 9(7)V99.
03 D-DRFT PIC 9(7)V99.
03 FILLER PIC X(17).

* FD PRINTF LABEL RECORD IS OMITTED.

01 P-REC PIC X(132).

* WORKING-STORAGE SECTION.

01 HD1.

03 FILLER PIC X(57) VALUE SPACES.
03 FILLER PIC X(15) VALUE "COMMERCIAL BANK".
03 FILLER PIC X(60) VALUE SPACES.

*

01 HD2.

03 FILLER PIC X(11) VALUE " DATE : ".
03 PDY PIC 99.
03 FILLER PIC X VALUE "/".
03 PMT PIC 99.
03 FILLER PIC XXX VALUE "/19".
03 PYR PIC 99.
03 FILLER PIC X(31) VALUE SPACES.
03 FILLER PIC X(23) VALUE
"REPORT FOR NEW BALANCE".
03 FILLER PIC X(34) VALUE SPACES.
03 FILLER PIC X(6) VALUE "PAGE: ".
03 PPAGE PIC ZZ9.
03 FILLER PIC X(11) VALUE SPACES.

*

01 HD3.

03 FILLER PIC X(51) VALUE
" ACCOUNT NO ACCOUNT NAME ".
03 FILLER PIC X(42) VALUE
"LAST BALANCE DRAFT AMOUNT".
03 FILLER PIC X(14) VALUE
"DEPOSIT AMOUNT".
03 FILLER PIC X(25) VALUE "NEW BALANCE".

*

01 DATAL.

03 FILLER PIC X(05) VALUE SPACES.
03 P-ACNO PIC 9(5).
03 FILLER PIC X(04) VALUE SPACES.
03 P-ACNAME PIC X(30).
03 FILLER PIC X(04) VALUE SPACES.
03 P-ENDBAL PIC Z(5)99.99.
03 FILLER PIC X(15) VALUE SPACES.
03 P-DRFT PIC Z(5)99.99.
03 FILLER PIC X(07) VALUE SPACES.

```

03 P-DPST PIC Z(5)99,99.
03 FILLER PIC X(10) VALUE SPACES.
03 P-NBAL PIC ++++++99,99.
03 FILLER PIC X(11) VALUE SPACES.
*
01 TOTL1.
03 FILLER PIC X(46) VALUE SPACES.
03 FILLER PIC X(27) VALUE
  "TOTAL OF ACTIVE RECORDS :".
03 T-ACTV PIC ZZ9.
03 FILLER PIC X(55) VALUE SPACE.
*
01 TOTL2.
03 FILLER PIC X(46) VALUE SPACES.
03 FILLER PIC X(27) VALUE
  "TOTAL OF LAST BALANCE :".
03 T-ENDBAL PIC Z(6)99,99.
03 FILLER PIC X(48) VALUE SPACES.
*
01 TOTL3.
03 FILLER PIC X(46) VALUE SPACES.
03 FILLER PIC X(27) VALUE
  "TOTAL OF DRAFT AMOUNTS :".
03 T-DRFT PIC Z(6)99,99.
03 FILLER PIC X(55) VALUE SPACE.
*
01 TOTL4.
03 FILLER PIC X(46) VALUE SPACES.
03 FILLER PIC X(27) VALUE
  "TOTAL OF DEPOSIT AMOUNTS :".
03 T-DPST PIC Z(6)99,99.
03 FILLER PIC X(48) VALUE SPACES.
01 TOTL5.
03 FILLER PIC X(46) VALUE SPACES.
03 FILLER PIC X(27) VALUE
  "TOTAL OF NEW BALANCES :".
03 T-NBAL PIC ++++++99,99.
03 FILLER PIC X(47).
*
* عدد ذات لرحم الجـالـجـ الجـالـجـ
*
77 C-NBAL PIC 9(8)V99 VALUE ZEROS.
77 C-ACTV PIC 9(4) VALUE ZEROS.
77 C-ENDBAL PIC 9(8)V99 VALUE ZEROS.
77 C-DPST PIC 9(8)V99 VALUE ZEROS.
77 C-DRFT PIC 9(8)V99 VALUE ZEROS.
*
* عدد ذات لرحم الجـالـجـ الجـالـجـ
*
77 LN-CNTR PIC 99 VALUE ZERO.
77 PG-CNTR PIC 999 VALUE ZERO.

```

```

■
* مساحات عمل لقبول التاريخ من الحساب
*
01 WDATE.
   03 WYR PIC 99.
   03 WMT PIC 99.
   03 WDY PIC 99.
*
* مساحه عمل من اجل حساب الرصيد
*
77 D-NBAL PIC S9(7)V99.
■
PROCEDURE DIVISION.
OPEN-FILES.
   OPEN      INPUT  DEPFIL
             OUTPUT PRINTF.
*
* قبول التاريخ وتحويل اليوم والشهر والسنة الى
* الى سطر العناوين
*
ACCEPT WDATE FROM DATE.
MOVE WYR TO PYR.
MOVE WMT TO PMT.
MOVE WDY TO PDY.
PERFORM N-PAGE THRU E-PAGE.
*
* قراءه سجل من الملف ثم المعالجة
*
READ-A-FILE.
   READ DEPFIL AT END GO TO END-OF-JOB.
* اختيار فيها اذن السجل وهل ام لا ؟
   IF D-CODE = "D" GO TO READ-A-FILE.
■
*
* حساب الرصيد الجديد
*
COMPUTE D-NBAL ROUNDED =
      D-ENDBAL + D-DPST - D-DRFT.
*
* تحويل الحقول من السجل الاعداد الى سجل الطباعة
* وكذلك نقل الرصيد الجديد الى محسوب
*
MOVE D-ACNO TO P-ACNO.
MOVE D-ACNAME TO P-ACNAME.
MOVE D-ENDBAL TO P-ENDBAL.
MOVE D-DRFT TO P-DRFT.
MOVE D-DPST TO P-DPST.
MOVE D-NBAL TO P-NBAL.
*
* اختيار نهاية الصفحة
*
IF LN-CNTR > 22
   PERFORM N-PAGE THRU E-PAGE.

```



```

* طباعة سجل على ملف للطابعة
*
      WRITE P-REC      FROM DATAL
                        AFTER 2.
      ADD 1            TO LN-CNTR.
*
* تحميل القديم المطلوب طباعتها في نهاية التقرير
*
      ADD 1            TO C-ACTV.
      ADD D-ENDBAL     TO C-ENDBAL.
      ADD D-DPST       TO C-DPST.
      ADD D-DRFT       TO C-DRFT.
      ADD D-NBAL       TO C-NBAL.
*
* بعد الانتهاء من معالجة السجل
* نذهب لمعالجة سجل آخر
*
      GO TO READ-A-FILE.
*
* لقراءة التالية هي لقب الصفحة وطباعة العنوانين
*
N-PAGE.
      MOVE ZERO        TO LN-CNTR.
      ADD 1            TO PG-CNTR.
      MOVE PG-CNTR     TO PPAGE.
      MOVE SPACES      TO P-REC.
      WRITE P-REC      FROM HD1 AFTER PAGE.
      MOVE SPACES      TO P-REC.
      WRITE P-REC      FROM HD2 AFTER 2 LINES.
      MOVE SPACES      TO P-REC.
      WRITE P-REC      FROM HD3 AFTER 3 LINES.
      MOVE SPACES      TO P-REC.
E-PAGE.
      EXIT.
*
* لقراءة التالية هي لطباعة المجموع النهائية
*
END-OF-JOB.
      MOVE C-ACTV      TO T-ACTV.
      MOVE C-ENDBAL    TO T-ENDBAL.
      MOVE C-DRFT      TO T-DRFT.
      MOVE C-DPST      TO T-DPST.
      MOVE C-NBAL      TO T-NBAL.
*
      WRITE P-REC      FROM SPACE AFTER PAGE.
      WRITE P-REC      FROM TOTL1 AFTER 32.
      WRITE P-REC      FROM TOTL2 AFTER 2.
      WRITE P-REC      FROM TOTL3 AFTER 2.
      WRITE P-REC      FROM TOTL4 AFTER 2.
      WRITE P-REC      FROM TOTL5 AFTER 2.
*
* نهاية العمل وإغلاق الملفات
*
      CLOSE DEFFIL, PRINTF.
      STOP RUN.

```

أنظر شكل المخرجات صفحة ٦٠١.

* حل تمرين رقم ٢ الفصل ١٣ -٥

IDENTIFICATION DIVISION.

PROGRAM-ID. TXT19.

AUTHOR. KHAYAT-JNAID.

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER. IBM-370.

OBJECT-COMPUTER. IBM-370.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT VALFILE ASSIGN TO UT-S-SYSIN.

SELECT PRINTFL ASSIGN TO UT-S-PRF.

*

DATA DIVISION.

FILE SECTION.

FD VALFILE RECORD CONTAINS 80 CHARACTERS,
BLOCK CONTAINS 0 RECORDS
LABEL RECORD IS STANDARD.

01 VAL-REC.

03 VAL-NO PIC 9(5).

03 VAL-NAME PIC X(20).

03 VAL-ADRESS PIC X(24).

03 VAL-DATE PIC 9(8).

03 VAL-CODE PIC 99.

03 VAL-AREA PIC 99.

03 VAL-BLNK PIC X(19).

*

FD PRINTFL RECORD CONTAINS 132 CHARACTERS
LABEL RECORD IS OMITTED.

01 P-REC PIC X(132).

*

WORKING-STORAGE SECTION.

01 HDR1.

03 FILLER PIC X(29) VALUE SPACES.

03 FILLER PIC X(8) VALUE 'I CODE I'.

03 FILLER PIC X(16) VALUE
' AREA NAME I'.

03 FILLER PIC X(20) VALUE
' SUBSCRIBER NO. I'.

03 FILLER PIC X(60) VALUE SPACES.

```

*
01 DLN.
03 FILLER PIC X(29) VALUE SPACES.
03 FILLER PIC X VALUE '|'.
03 FILLER PIC XX VALUE SPACES.
03 P-CODE PIC Z9.
03 FILLER PIC XX VALUE SPACES.
03 FILLER PIC X VALUE '|'.
03 FILLER PIC XX VALUE SPACES.
03 P-AREA PIC X(10).
03 FILLER PIC XXX VALUE SPACES.
03 FILLER PIC X VALUE '|'.
03 FILLER PIC X(9) VALUE SPACES.
03 P-CNTR PIC ZZ9.
03 FILLER PIC X(4) VALUE SPACES.
03 FILLER PIC X(8) VALUE ' | '.
03 FILLER PIC X(50) VALUE SPACES.

*
01 DASHES.
03 FILLER PIC X(29) VALUE SPACES.
03 FILLER PIC X(44) VALUE ALL '-'.
03 FILLER PIC X(30) VALUE SPACES.

*
01 TOTL.
03 FILLER PIC X(29) VALUE SPACES.
03 FILLER PIC X VALUE '|'.
03 FILLER PIC X(14) VALUE SPACES.
03 FILLER PIC X(17) VALUE ' OVERALL TOTAL:'.
03 POVALL PIC ZZ99.
03 FILLER PIC X(04) VALUE SPACES.
03 FILLER PIC X(8) VALUE ' | '.
03 FILLER PIC X(50) VALUE SPACES.

01 TBL-AREA.
03 FILLER PIC X(15) VALUE
    '01RIYADH 000'.
03 FILLER PIC X(15) VALUE
    '02JEDDAH 000'.
03 FILLER PIC X(15) VALUE
    '03DAMMAM 000'.
03 FILLER PIC X(15) VALUE
    '04QUASEEM 000'.

```

```

03 FILLER PIC X(15) VALUE
    '05HAYEL      000'.
03 FILLER PIC X(15) VALUE
    '06ALMADINA  000'.
03 FILLER PIC X(15) VALUE
    '07ABHA      000'.
03 FILLER PIC X(15) VALUE
    '08BALGURASHI000'.
03 FILLER PIC X(15) VALUE
    '09ASEER     000'.
03 FILLER PIC X(15) VALUE
    '10SHAQRA    000'.

```

*

```

01  RED-TBL-AREA REDEFINES TBL-AREA.
    02 TBL-ITM OCCURS 10 TIMES ASCENDING
        KEY TBL-CODE INDEXED BY IX.
        03 TBL-CODE PIC 99.
        03 TBL-AREA PIC X(10).
        03 TBL-CNTR PIC 999.

```

*

```

77  OVAL-CNTR          PIC 9(4) VALUE ZERO.
    PROCEDURE DIVISION.
    STRT.
        OPEN INPUT VALFILE OUTPUT PRINTFL.
    RD-RC.
        READ VALFILE AT END GO TO END-PROCESS.
        SEARCH ALL TBL-ITM AT END GO TO RD-RC
            WHEN VAL-AREA = TBL-CODE(IX)
                ADD 1 TO TBL-CNTR(IX).
        ADD 1 TO OVAL-CNTR.
        GO TO RD-RC.

```

*

```

END-PROCESS.
    WRITE P-REC AFTER PAGE.
    WRITE P-REC FROM DASHES AFTER 10.
    MOVE SPACES TO P-REC.
    WRITE P-REC FROM HDR1.
    MOVE SPACES TO P-REC.
    WRITE P-REC FROM DASHES.
    PERFORM LAST-PRINT THRU EXTP
        VARYING IX FROM 1 BY 1
            UNTIL IX > 10.

```

MOVE OVALL-CNTR TO POVALL.
 WRITE P-REC FROM TOTL.
 WRITE P-REC FROM DASHES.
 CLOSE VALFILE PRINTFL.
 STOP RUN.

LAST-PRINT.

MOVE TBL-CODE(IX) TO P-CODE.
 MOVE TBL-AREA(IX) TO P-AREA.
 MOVE TBL-CNTR(IX) TO P-CNTR.
 MOVE SPACES TO P-REC.
 WRITE P-REC FROM DLN AFTER 1.
 MOVE SPACES TO P-REC.
 WRITE P-REC FROM DASHES AFTER 1.

EXTP.

EXIT.

شكل المخرجات

| CODE | AREA NAME | SUBSCRIBER NO. |
|-----------------|------------|----------------|
| 1 | RIYADH | 5 |
| 2 | JEDDAH | 2 |
| 3 | DAMMAM | 4 |
| 4 | QUASEEM | 3 |
| 5 | HAYEL | 3 |
| 6 | ALMADINA | 4 |
| 7 | ABHA | 2 |
| 8 | BALGURASHI | 3 |
| 9 | ASEER | 0 |
| 10 | SHAQRA | 3 |
| OVERALL TOTAL : | | 29 |

* حل تمرين رقم ٣ في الفصل ١٣-٥ *

IDENTIFICATION DIVISION.
PROGRAM-ID. TXT15.
AUTHOR. KHAYAT-JNAID.

*

* يقوم هذا البرنامج بمعالجة ملف مقرر

*

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.
SPECIAL-NAMES. DECIMAL-POINT IS COMMA.
INPUT-OUTPUT SECTION.
FILE-CONTROL.

SELECT PAYMST ASSIGN TO PAYF
ORGANIZATION IS INDEXED;
ACCESS MODE SEQUENTIAL;
RECORD KEY EMP-NO.
SELECT P-FILE ASSIGN TO UT-S-PRFL.

DATA DIVISION.

FILE SECTION.

FD PAYMST LABEL RECORD IS STANDARD
DATA RECORD IS PAY-REC.

01 PAY-REC.

03 EMP-NO PIC X(5).
03 EMP-NAME PIC X(30).
03 EMP-SAL PIC 9(5)V99.
03 EMP-HRS1 PIC 9(3).
03 EMP-HRS2 PIC 9(3).
03 EMP-HRS3 PIC 9(3).
03 EMP-OVTM PIC 9(5)V99.
03 FILLER PIC X(22).

*

FD P-FILE LABEL RECORD IS OMITTED
DATA RECORD IS P-REC.

01 P-REC PIC X(132).

*

WORKING-STORAGE SECTION.

01 HDR-1.
03 FILLER PIC X(56) VALUE SPACES.
03 FILLER PIC X(19) VALUE

| | | | |
|----|--------|-----|---------------------|
| | | | 'شركة الليبن' |
| 03 | FILLER | PIC | X(57) VALUE SPACES. |
| 01 | HDR-2. | | |
| 03 | FILLER | PIC | X(20) VALUE |
| | | | 'رقم الموقت' |
| 03 | FILLER | PIC | X(30) VALUE |
| | | | 'الاسم' |
| 03 | FILLER | PIC | X(30) VALUE |
| | | | 'الراتب' |
| 03 | FILLER | PIC | X(52) VALUE |
| | | | 'ملاحظات' |

*

| | | | |
|----|------------|-----|---------------------|
| 01 | DATA-LINE. | | |
| 03 | FILLER | PIC | X(02) VALUE SPACES. |
| 03 | P-NO | PIC | ZZZ99. |
| 03 | FILLER | PIC | X(08) VALUE SPACES. |
| 03 | P-NAME | PIC | X(30). |
| 03 | FILLER | PIC | X(04) VALUE SPACES. |
| 03 | P-SAL | PIC | Z(4)9,99. |
| 03 | FILLER | PIC | X(06) VALUE SPACES. |
| 03 | P-OVTM | PIC | ZZZ99,99. |
| 03 | FILLER | PIC | X(09) VALUE SPACES. |
| 03 | P-COMN | PIC | X(16). |
| 03 | FILLER | PIC | X(10) VALUE SPACES. |

*

| | | | |
|----|-----------|-----|---------------------|
| 01 | TOTAL-L1. | | |
| 03 | FILLER | PIC | X(39) VALUE SPACES. |
| 03 | PCNTR1 | PIC | Z(5)9,99. |
| 03 | FILLER | PIC | X(84) VALUE |
| | | | 'مجموع الاجمور' |
| 01 | TOTAL-L2. | | |
| 03 | FILLER | PIC | X(39) VALUE SPACES. |
| 03 | PCNTR2 | PIC | Z(5)9,99. |
| 03 | FILLER | PIC | X(84) VALUE |
| | | | 'مجموع خارج الدوام' |
| 01 | TOTAL-L3. | | |
| 03 | FILLER | PIC | X(39) VALUE SPACES. |
| 03 | PCNTR3 | PIC | Z(5)9,99. |
| 03 | FILLER | PIC | X(84) VALUE |
| | | | 'عدد العمال الكلي' |

```

*
* عدادات تستخدم للتجميع
01  COUNTERS.
      03 CNTR1 PIC      9(6)V99.
      03 CNTR2 PIC      9(6)V99.
      03 CNTR3 PIC      9999.
77  HALF-SAL PIC      9(5)V99.
77  HOURS-RAT PIC      9(3)V99.
77  E-O-J PIC      XXX VALUE 'NO'.
77  PEMK PIC      X(16) VALUE
      'تجاوز الحد الاعلى'.

*
PROCEDURE DIVISION.
      OPEN      I-O      PAYMST
              OUTPUT P-FILE.
* اعطاء القيمة صفر للعدادات
      MOVE      ZEROS      TO      COUNTERS.
      WRITE     P-REC AFTER PAGE.
      WRITE     P-REC FROM HDR-1.
      WRITE     P-REC FROM HDR-2 AFTER 2.
      PERFORM   READ-FILE.
* امل الاساسي للبرنامج
      MAIN-LOOP.
          PERFORM READ-RTN THRU EXT-RTN
              UNTIL E-O-J = 'YES'.
          PERFORM FINAL-RTN.
          CLOSE  PAYMST P-FILE.
          STOP  RUN.
      READ-RTN.

*
* حساب اجره الساعة
      COMPUTE HOURS-RAT ROUNDED =
          EMP-SAL / ( 30 * 8 ).
* حساب خارج الدوام
      COMPUTE EMP-OVTH ROUNDED =
          HOURS-RAT * ( EMP-HRS1 +
              EMP-HRS2 * 1,5 +
              EMP-HRS3 * 2 ).
* حساب نصف الدوام
      DIVIDE EMP-SAL BY 2 GIVING HALF-SAL.

```



```

*
* : مقارنة نصف الراتب مع خارج الدوام
  IF EMP-OVTH GREATER THAN HALF-SAL
      MOVE HALF-SAL TO EMP-OVTH.
  PERFORM PRINT-LINE.
  ADD EMP-SAL TO CNTR1.
  ADD EMP-OVTH TO CNTR2.
  ADD 1 TO CNTR3.
* إعادة كتابته السجل على نفس الملف
* المخصص بعد تعديل حقل خارج الدوام
  REWRITE PAY-REC.
* قراءته سجل جديد من الملف المخصص:
*
  READ-FILE.
  READ PAYMST NEXT
  AT END MOVE 'YES' TO E-O-J.
*
  EXT-RTN.
  EXIT.
*
  PRINT-LINE.
* تحريك المقول والنتائج الى سطر الطباعة:
  MOVE EMP-NO TO P-NO.
  MOVE EMP-NAME TO P-NAME.
  MOVE EMP-SAL TO P-SAL.
  MOVE EMP-OVTH TO P-OVTH.
*
* : كتابته السجل على ملف الطباعة
  WRITE P-REC FROM
  DATA-LINE AFTER ADVANCING 2 LINES.
  EXT-P.
  EXIT.
*
* : طباعة أسطر المجاميع النهائية
  FINAL-RTN.
  MOVE CNTR1 TO PCNTR1.
  WRITE P-REC FROM TOTAL-L1.
  MOVE CNTR2 TO PCNTR2.
  WRITE P-REC FROM TOTAL-L2.
  MOVE CNTR3 TO PCNTR3.
  WRITE P-REC FROM TOTAL-L3.
  EXITF.
  EXIT.

```

* O-13 في الفصل O ●
IDENTIFICATION DIVISION.
PROGRAM-ID. TXT17.
AUTHOR. KHAYAT-JNAID.
*
* يؤوم هذا البرنامج بعمل تدقق لملف تعديل
* مع الملف الاساسي ويطلع السجل الذي ليس له اصل
*
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
 SELECT SUBFILE ASSIGN TO UT-S-SFIL.
 SELECT INNFILE ASSIGN TO UT-S-INFL.
 SELECT CONFIL ASSIGN TO UT-S-CONF.
 SELECT PRINTFL ASSIGN TO UT-S-PFLL.
DATA DIVISION.
FILE SECTION.
*
* توضيح الملف الاساسي :
FD SUBFILE LABEL RECORD IS STANDARD
 BLOCK CONTAINS 0 RECORDS
 DATA RECORD IS SUB-REC.
01 SUB-REC.
 03 SUB-NO PIC X(5).
 03 SUB-NAME PIC X(20).
 03 SUB-ADRESS PIC' X(24).
 03 SUB-DATE.
 05 SUB-YEAR PIC 9(4).
 05 SUB-MNTH PIC 9(2).
 05 SUB-DAY PIC 9(2).
 03 SUB-CODE PIC 99.
 03 SUB-AREA PIC 99.
 03 SUB-CNTR PIC 9(7).
 03 SUB-LREAD PIC 9(7).
 03 FILLER PIC X(5).
*
FD INNFILE LABEL RECORD IS STANDARD
 BLOCK CONTAINS 0 RECORDS
 DATA RECORD IS IN-REC.

```

01  IN-REC.
    03  IN-NO      PIC  X(5).
    03  IN-READ    PIC  9(7).
    03  IN-DATE    PIC  9(8).
    03  IN-COMNT   PIC  X(30).
    03  FILLER     PIC  X(30).

*
FD  CONFILE LABEL RECORD IS STANDARD
    BLOCK CONTAINS 0 RECORDS
    DATA RECORD IS CON-REC.

01  CON-REC.
    03  CON-NO     PIC  9(5).
    03  CON-READ   PIC  9(7).
    03  CON-DATE   PIC  9(8).
    03  CON-COMNT  PIC  X(30).
    03  FILLER     PIC  X(30).

*
* توصيف ملف الطباعة :
FD  PRINTFL LABEL RECORD IS OMITTED
    DATA RECORD IS P-REC.

01  P-REC      PIC  X(132).

*
WORKING-STORAGE SECTION.

*
01  HDR-1.
    03  FILLER PIC  X(58) VALUE SPACES.
    03  FILLER PIC  X(16) VALUE
        'شركة الكوربا'.
    03  FILLER PIC  X(58) VALUE SPACES.

*
01  HDR-2.
    03  DAY-P PIC  99.
    03  FILLER PIC  X VALUE '/'.
    03  MNTH-P PIC  99.
    03  FILLER PIC  XXX VALUE '/19'.
    03  YEAR-P PIC  99.
    03  FILLER PIC  X(12) VALUE
        'تاريخ :'.
    03  FILLER PIC  X(29) VALUE SPACES.
    03  FILLER PIC  X(38) VALUE
        'قائمة بالسجلات المرصودة'.
    03  FILLER PIC  X(29) VALUE SPACES.

```



```

*
PROCEDURE DIVISION.
    OPEN      INPUT  SUBFILE INNFILE
              OUTPUT PRINTFL CONFILE.

*
*      قبول التاريخ وطباعة العناوين
*      وقراءه السجل الاول من الملف
*
    ACCEPT    CURENT-DATE FROM DATE.
    MOVE      C-YEAR      TO   YEAR-P.
    MOVE      C-MNTH     TO   MNTH-P.
    MOVE      C-DAY      TO   DAY-P.
    PERFORM   NEW-PAGE    THRU EFTP.

*
*      المعالجة الرئيسية للبرنامج
*      MAIN-LOOP.
    PERFORM   READ-TRANS  THRU READ-MASTER.
    COMPAR.

*
*
*
    IF EDJOB = 'YES'
        CLOSE SUBFILE INNFILE
        CONFILE PRINTFL STOP RUN.
    IF IN-NO < SUB-NO PERFORM PRINT-LINE
        PERFORM READ-TRANS
        GO TO COMPAR

    ELSE
    IF SUB-NO < IN-NO PERFORM READ-MASTER
        GO TO COMPAR

    ELSE
    WRITE CON-REC FROM IN-REC.
    GO TO MAIN-LOOP.
*      قراءه سجل من ملف التعديل
    READ-TRANS.
    READ INNFILE AT END
    MOVE 'YES' TO EDJOB.

    EXIT.

    EXIT.
*      قراءه سجل من الملف الاساسي
    READ-MASTER.
    READ SUBFILE AT END

```

```

                                MOVE HIGH-VALUE TO SUB-NO.
EXTM.
  EXIT.
PRINT-LINE.
  MOVE IN-NO      TO P-NO.
  MOVE IN-READ    TO P-READ.
  MOVE IN-DATE    TO P-DATE.
  MOVE IN-COMNT   TO P-COMNT.
  IF LINE-CTR GREATER THAN 29
    PERFORM NEW-PAGE THRU EXTP.
  WRITE P-REC FROM DATA-LINE
    AFTER ADVANCING 2 LINES.
  ADD 1 TO LINE-CTR.
*   : فقره طلب الصفحة وطبع العناوين :
NEW-PAGE.
  MOVE SPACES      TO P-REC.
  WRITE P-REC      AFTER PAGE.
  MOVE ZERO        TO LINE-CTR.
  ADD 1           TO PAGE-CTR.
  MOVE PAGE-CTR    TO P-CTR.
  WRITE P-REC FROM HDR-1 AFTER 1 LINE.
  WRITE P-REC FROM HDR-2 AFTER 1 LINE.
  WRITE P-REC FROM HDR-3 AFTER 1 LINE.
EXTP.
  EXIT.
*
```

أنظر شكل المخرجات صفحة ٦٠١



نتيجة البرنامج

LOCATION AGE

| | |
|--------|----|
| RIYADH | 27 |
| RIYADH | 29 |
| RIYADH | 24 |
| RIYADH | 27 |
| RIYADH | 28 |
| RIYADH | 23 |
| RIYADH | 33 |
| RIYADH | 32 |
| RIYADH | 30 |
| RIYADH | 33 |
| RIYADH | 33 |
| RIYADH | 33 |
| RIYADH | 33 |
| RIYADH | 33 |

=====

| NAME | SALARY |
|------|--------|
|------|--------|

=====

| | |
|-----------------|------|
| ZUHAIR KHAYAT | 2500 |
| ALI ALGHANDI | 2000 |
| SALEH MONAJED | 3000 |
| ABDULLAH GHANEM | 2500 |
| ALI SALEH | 3000 |
| FERAS A MAD | 2000 |
| KHALDOON JNAID | 2000 |
| SALEM A:DULLA | 2500 |
| IBRAHIM ALHARBI | 2500 |
| IBRAHIM AOTAIBI | 2400 |
| SABER ALYOUSEF | 2400 |
| HAYTHAM ALZOUHD | 3000 |
| ISA ALMADANI | 3000 |
| KHALED HJAIN | 3000 |

قائمة اذطاع تعديل

| اسم | الملاحظات |
|-------------|--------------|
| RAJAB | جل غير موجود |
| AHMED JNAID | جل غير موجود |
| AHMED | جل غير موجود |
| KHANAWATI | جل غير موجود |

الملف الاساسي

| رقم المشترك | المشترك |
|-------------|----------|
| 10008 | SUBHI |
| 10017 | FERAS |
| 10018 | KHALDOON |
| 10023 | SALEM |

- عدد السجلات المضافة 25
- عدد السجلات المعدله 3
- عدد السجلات المحذوفه 0
- عدد السجلات المرفوضه 4

| SUBSCRIPTION DATE | SUBS. CODE | SUBS. AREA | COUNTR | LAST RD |
|-------------------|------------|------------|--------|---------|
| 11/12/1404 | 01 | 01 | 1233 | 100 |
| 11/12/1404 | 01 | 01 | 1234 | 100 |
| 11/12/1404 | 02 | 01 | 1235 | 0 |
| 12/01/1404 | 02 | 01 | 1236 | 200 |
| 12/01/1404 | 02 | 01 | 1237 | 0 |
| 12/01/1404 | 03 | 01 | 1238 | 300 |
| 12/01/1404 | 03 | 01 | 1240 | 400 |
| 12/01/1404 | 01 | 01 | 1242 | 500 |
| 02/01/1405 | 01 | 01 | 1243 | 0 |
| 02/01/1405 | 04 | 01 | 1244 | 700 |
| 02/01/1405 | 04 | 01 | 1245 | 0 |
| 02/01/1405 | 04 | 01 | 1200 | 0 |
| 02/01/1405 | 05 | 01 | 1203 | 0 |
| 02/01/1405 | 01 | 01 | 1205 | 0 |
| 02/01/1405 | 05 | 01 | 1204 | 0 |
| 10/01/1405 | 06 | 01 | 1207 | 0 |
| 02/01/1405 | 02 | 01 | 1208 | 200 |
| 02/02/1405 | 07 | 01 | 1209 | 200 |
| 02/02/1405 | 07 | 01 | 1213 | 100 |
| 02/03/1405 | 06 | 01 | 1333 | 900 |
| 02/03/1405 | 02 | 01 | 1343 | 800 |
| 02/03/1405 | 03 | 01 | 1453 | 700 |
| 02/03/1405 | 03 | 01 | 1563 | 100 |
| 02/03/1405 | 08 | 01 | 1673 | 600 |
| 02/04/1405 | 08 | 01 | 983 | 400 |
| 02/04/1405 | 09 | 01 | 1783 | 200 |
| 02/04/1405 | 01 | 01 | 1903 | 200 |
| 02/04/1405 | 02 | 01 | 1913 | 200 |
| 02/04/1405 | 02 | 01 | 1983 | 200 |

ELECTRIC COMPANY
LIST OF SUBSCRIBER RECORDS

DATE: 21/11/1984

| SUBSCRIBER NO. | SUBS. NAME | SUBS. ADDRESS |
|----------------|--------------------|-----------------------|
| 10004 | ZUHAIR KHAYAT | RYIADH P.O BOX 112233 |
| 10005 | SAHI ALHALABI | RYIADH P.O BOX 2066 |
| 10006 | MOJAHED ALHARBI | RYIADH P.O BOX 230 |
| 10007 | BADR ALJUNDI | RYIADH P.O BOX 1102 |
| 10008 | SUBHI RAJAB | RYIADH P.O BOX 102 |
| 10009 | AHMAD HAMEED GHAZI | RYIADH P.O BOX 1002 |
| 10017 | FERAS AHMED JNAID | RYIADH P.O BOX 110 |
| 10018 | KHALDOON AHMED | RYIADH P.O BOX 1202 |
| 10019 | SAMEER HUSNI | RYIADH P.O BOX 1103 |
| 10021 | MANSUR AL GHADAN | RYIADH P.O BOX 1105 |
| 10022 | ABDULAZIZ QUAIZ | RYIADH P.O BOX 1023 |
| 10023 | SALEM KHANAWATI | RYIADH P.O BOX 1103 |
| 10024 | BASHIR ZUNDAH | RYIADH P.O BOX 210 |
| 10027 | ALI AL YUSEF | RYIADH P.O BOX 230 |
| 10029 | MOHAMED ZAKI | RYIADH P.O BOX 103 |
| 10033 | MDLHAM ZAINI | RYIADH P.O BOX 450 |
| 10034 | AHMED JAMALI | RYIADH P.O BOX 44 |
| 10036 | ABDULRAHMAN HARBI | RYIADH P.O BOX 11 |
| 10038 | ABDUL AZIZ MANSOOR | RYIADH P.O BOX 4567 |
| 10039 | GHAS ALCHATTA | RYIADH P.O BOX 234 |
| 10048 | HUSNI AL SHHEMRI | RYIADH P.O BOX 556 |
| 10049 | MONEER QUATARI | RYIADH P.O BOX 678 |
| 10050 | MOHAMED ABD ALAZIZ | RYIADH P.O BOX 45 |
| 10051 | NASER AL DHIBI | RYIADH P.O BOX 32 |
| 10052 | OTHMAN AL BASHEER | RYIADH P.O BOX 1443 |
| 10053 | ALMAJED YANYAMI | RYIADH P.O BOX 1244 |
| 10056 | MAAMAR MAKID | RYIADH P.O BOX 1223 |
| 10058 | LINAH KHAYAT | RYIADH P.O BOX 1026 |
| 10059 | YOUSUF QABOLI | RYIADH P.O BOX 1003 |

DATE: 21/11/1984

| SOCIAL SECURITY | SOCIAL-FUND | NET PAY | SIGNATURE |
|-----------------|-------------|---------|-----------|
| 250,00 | 190,00 | 5160,00 | |
| 200,00 | 140,00 | 4060,00 | |
| 300,00 | 240,00 | 5960,00 | |
| 200,00 | 140,00 | 4060,00 | |
| 65,00 | 19,00 | 1616,00 | |
| 115,00 | 55,00 | 2530,00 | |
| 220,00 | 160,00 | 4420,00 | |
| 275,00 | 215,00 | 5410,00 | |
| 50,00 | 10,00 | 1340,00 | |
| 50,00 | 10,00 | 1340,00 | |
| 115,00 | 55,00 | 2530,00 | |
| 225,00 | 165,00 | 4510,00 | |
| 195,00 | 135,00 | 3970,00 | |
| 335,00 | 275,00 | 6490,00 | |
| 115,00 | 55,00 | 2530,00 | |
| 45,00 | 9,00 | 1246,00 | |
| 45,00 | 9,00 | 1246,00 | |
| 40,00 | 8,00 | 1152,00 | |
| 170,00 | 110,00 | 3520,00 | |
| 225,00 | 165,00 | 4610,00 | |
| 35,00 | 7,00 | 1158,00 | |
| 225,00 | 165,00 | 4610,00 | |
| 225,00 | 165,00 | 4610,00 | |
| 170,00 | 110,00 | 3620,00 | |
| 100,00 | 40,00 | 2360,00 | |
| 115,00 | 55,00 | 2630,00 | |
| 100,00 | 40,00 | 2360,00 | |
| 75,00 | 25,00 | 1900,00 | |
| 70,00 | 22,00 | 1808,00 | |

BRIGHT STAR COMPANY
PAYROLL REPORT

PAGE NO : 1

| EMPLOYEE NO. | EMPLOYEE NAME | BASE SALARY | TRANSPORTATION |
|--------------|--------------------|-------------|----------------|
| 1004 | ZUHAIR KHAYAT | 5000,00 | 600,00 |
| 1005 | SAMI ALHALABI | 4000,00 | 400,00 |
| 1006 | MOJAHED ALHARDI | 6000,00 | 500,00 |
| 1007 | BADR ALJUNDI | 4000,00 | 400,00 |
| 1008 | SUBHI RAJAB | 1300,00 | 400,00 |
| 1009 | AHMAD HAMEED GHAZI | 2300,00 | 400,00 |
| 1017 | FERAS AHMED JNAID | 4400,00 | 400,00 |
| 1018 | KHALDOON AHMED | 5500,00 | 400,00 |
| 1019 | SAMEER HUSNI | 1000,00 | 400,00 |
| 1021 | HANSUR AL GHADAN | 1000,00 | 400,00 |
| 1022 | ABDULAZIZ QUATZ | 2300,00 | 400,00 |
| 1023 | SALEM KHANAWATI | 4500,00 | 400,00 |
| 1024 | DASHIR ZUNDAN | 3900,00 | 400,00 |
| 1027 | ALI AL YUSEF | 6700,00 | 400,00 |
| 1029 | MOHAMED ZAKI | 2300,00 | 400,00 |
| 1033 | MOLHAN ZAINI | 900,00 | 400,00 |
| 1034 | AHMED JAMALI | 900,00 | 400,00 |
| 1036 | ABDULRAHMAN HARDI | 800,00 | 400,00 |
| 1038 | ABDUL AZIZ HANSOOR | 3400,00 | 400,00 |
| 1039 | GHIAS ALCHATTA | 4500,00 | 500,00 |
| 1048 | HUSNI AL SHMEMRI | 700,00 | 500,00 |
| 1049 | MONEER QUATARI | 4500,00 | 500,00 |
| 1050 | MOHAMED ABD ALAZIZ | 4500,00 | 500,00 |
| 1051 | NASER AL DHIBI | 3400,00 | 500,00 |
| 1052 | OTIHMAN AL DASNEER | 2000,00 | 500,00 |
| 1053 | ALMAJED YAHYAMI | 2300,00 | 500,00 |
| 1056 | HAAMAR MAKID | 2000,00 | 500,00 |
| 1058 | LINAH KHAYAT | 1500,00 | 500,00 |
| 1059 | YOUSUF QABOLI | 1400,00 | 500,00 |

شركة الكورباء

قائمة بالمشتريين والمستلزمات
لشهر ١٤٠٥ هـ

رقم :

| رقم المنطقة | عنوان المشتري | اسم المشتري | المشتري |
|-------------|----------------------|-------------------|---------|
| ٠١ | RYIADH P,O BOX ١١٢٣٣ | ZUHAIR KHAYAT | ١٠ |
| ٠١ | RYIADH P,O BOX ٢٠٦٦ | SAMI ALHALABI | ١٠ |
| ٠١ | RYIADH P,O BOX ٢٣٠ | MOJAHED ALHARBI | ١٠ |
| ٠١ | RYIADH P,O BOX ١١٠٢ | BADR ALJUNDI | ١٠ |
| ٠١ | RYIADH P,O BOX ١١٠ | FERAS AHMED JNAID | ١٠ |
| ٠١ | RYIADH P,O BOX ١١٠٣ | SAMEER HUSNI | ١٠ |
| ٠١ | RYIADH P,O BOX ٤٥٠ | MOLHAM ZAINI | ١٠ |
| ٠١ | RYIADH P,O BOX ٣٢ | NASER AL OHIBI | ١٠ |
| ٠١ | RYIADH P,O BOX ١٤٤٣ | OTHMAN AL BASHEER | ١٠ |
| ٠١ | RYIADH P,O BOX ١٢٤٤ | ALMAJED YAHYAWI | ١٠ |
| ٠١ | RYIADH P,O BOX ١٢٢٣ | MAAMAR MAKID | ١٠ |
| ٠١ | RYIADH P,O BOX ١٠٢٦ | LINAH KHAYAT | ١٠ |
| ٠١ | RYIADH P,O BOX ١٠٠٣ | YOUSUF QABOLI | ١٠ |

تاريخ: ١٩٨٤/١١/٢١

| رمز الاستهلاك رقم | العدد | مقدار الاستهلاك | المستحق للدفع |
|-------------------|-------|-----------------|---------------|
| ٠١ | ١٦٣٣ | ٦٣٠ | ١١,٦٤ |
| ٠١ | ١٦٣٤ | ٥٣٠ | ١٩,٦٨ |
| ٠٢ | ١٦٣٥ | ١٥٥٠ | ٥٦,٤٠ |
| ٠٢ | ١٦٣٦ | ٣٣٠ | ١٦,٤٤ |
| ٠٣ | ١٦٤٠ | ٢٥٤٥ | ٩٢,٦٢ |
| ٠١ | ١٦٤٣ | ١٦٧٠ | ٦٠,٧٢ |
| ٠٦ | ١٦٠٧ | ١٤٥٥ | ٠,٦٠ |
| ٠٨ | ١٦٧٣ | ٤٥٤٦ | ١٠٩,٧٠ |
| ٠٨ | ٩٨٣ | ٢٢٢٢ | ٥٣,٩٢ |
| ٠٩ | ١٧٨٣ | ١٨٢ | ٤,٩٦ |
| ٠١ | ١٩٠٣ | ٧٥٢ | ٢٧,٦٧ |
| ٠٢ | ١٩١٣ | ٣٣٢ | ١٦,٥٣ |
| ٠٢ | ١٩٨٣ | ٤٥٢ | ٢٢,٢٩ |

الذاتية التشغيلية

عدد المستفيدين : ٠١٣
 مقدار الكيلو ووات المصروفة: ١٦٧٩٦
 مقدار المبالغ المستحقة: ٤٩٢,٧٧

PAGE NO : 1

SUBSCRIPTION DATE SUB. C O D E SUB. AREA LAST READING

| | | | |
|------------|----|----|-------|
| 1405/01/02 | 01 | 01 | 1243 |
| 1405/01/02 | 05 | 01 | 1203 |
| 1005/00/00 | 05 | 01 | 1203 |
| * * * | | | |
| 1405/01/02 | 05 | 00 | 61206 |
| | | * | |
| 0000/00/00 | 00 | 00 | 00 |
| * * * | * | * | * |

PAGE :

| ITEM PRICE | MINIMUM LEVEL | COMMENTS |
|------------|---------------|----------|
| 22.00 | 23 | REORDER |
| 100.00 | 25 | REORDER |
| 32.00 | 100 | |
| 2.00 | 100 | |
| 10.00 | 400 | REORDER |
| 3.00 | 100 | |
| 10.00 | 50 | |
| 1.00 | 200 | REORDER |
| 3.00 | 300 | REORDER |
| 2.00 | 100 | |
| 22.00 | 23 | REORDER |

ELECTRIC COMPANY

DATE: 21/11/1984

LIST OF INVALID SUBSCRIBER RECORDS

| SUBSCRIBER NO. | SUBS. NAME | SUBS. ADDRESS |
|----------------|----------------------|---------------------|
| 10019 | AYMAN KHALIFA | RYIADH P.O BOX 4103 |
| * | | |
| 10025 | YARUB JAMEEL | |
| | | * |
| 10026 | MJDI ABDALNASER | RYIADH P.O BOX 3333 |
| 10030 | MANSOOR ABDO | RYIADH P.O BOX 4103 |
| 10089 | 33333333333333333333 | |
| | * | * |

نتائج البرنامج رقم ٢ الفصل الخامس صفحة ١٩٨

S T O C K L I S T

DATE : 84/11/21

| ITEM NO. | ITEM DESCRIPTION | QUANTITY | ITEM CODE |
|----------|------------------|----------|-----------|
| 10012 | TABLES | 20 | 30 |
| 10034 | TELEPHONES | 23 | 40 |
| 10044 | CLASSES | 250 | 60 |
| 10054 | PENCILS | 125 | 50 |
| 10055 | NOTE BOOKS | 300 | 140 |
| 10056 | GLUE | 300 | 100 |
| 10060 | BOOKS | 200 | 80 |
| 10064 | CARTOON FILES | 163 | 120 |
| 10070 | PENTEL PEN | 220 | 130 |
| 10088 | PLASTIC FILES | 130 | 110 |
| 10104 | ARM CHAIRS | 23 | 10 |

نتائج برنامج رقم ٢ الفصل السابع صفحة ٢٤٩

| DRAFT AMOUNT | DEPOSIT AMOUNT | NEW BALANCE |
|--------------|----------------|-------------|
| 432,34 | 4530,00 | +27497,66 |
| 500,00 | 3000,00 | +35500,00 |
| 700,00 | 200,00 | +4060,00 |
| 3000,00 | 3000,00 | +6300,00 |
| 7000,00 | 2000,00 | -700,00 |
| 5000,00 | 10000,00 | +10550,00 |
| 00,00 | 2000,00 | +22000,00 |
| 2000,00 | 3000,00 | +7500,00 |
| 333,33 | 3300000,00 | +3300000,00 |
| 800000,00 | 00,00 | -793400,00 |
| 500000,00 | 4000,00 | -495600,00 |
| 00,00 | 3000,00 | +8500,00 |
| 300000,00 | 500000,00 | +200000,00 |

شركة الكهرباء
خاتمة بالسجلات المدخولة من ملف الاستاذك

محفظة رقم: 1

تاريخ الدفعة

ملاحظات

1405/02/02

COUNTER OUT

1405/02/02

COUNTER OUT

1405/02/02

COUNTER OUT

1405/02/02

COUNTER OUT

1405/02/02

COUNTER OUT

1405/02/02

COUNTER OUT

| ACCOUNT NO | ACCOUNT NAME | LAST BALANCE |
|------------|-----------------------|--------------|
| 20043 | ZUHAIR AL KHAYAT | 23400,00 |
| 20046 | FATHI IMAM AL KHAIR | 33000,00 |
| 20047 | FIRAS AHMED JNAID | 4560,00 |
| 20048 | LINAH AL KHAYAT | 6300,00 |
| 20049 | KHALDOON AHMED JUNAID | 4300,00 |
| 20050 | HUDA IBRAHIM ALCHATTA | 5550,00 |
| 20051 | SALMA ZUHAIR ALKHAYAT | 20000,00 |
| 20053 | JAWHARA AL MOJEL | 6500,00 |
| 20056 | IBRAHIM AL SANEH | 333,33 |
| 20057 | NASER AL OHEBI | 6600,00 |
| 20058 | MAHMOOD MOHAMED ALI | 400,00 |
| 20060 | MOHAMED ABO SALEM | 5500,00 |
| 20061 | NAJEEB BEN SAEID | 00,00 |

| | |
|---------------------------|-------------|
| TOTAL OF ACTIVE RECORDS : | 13 |
| TOTAL OF LAST BALANCE : | 116443,33 |
| TOTAL OF DRAFT AMOUNTS : | 1618965,67 |
| TOTAL OF DEPOSIT AMOUNTS: | 3834730,00 |
| TOTAL OF NEW BALANCES : | +2332207,66 |

نتائج برنامج التمارين العامة تمرين ٢ صفحة ٤٨٩

تاريخ: 21/11/1984

| رقم المشترك | مقدار القسط ٢٠٤٤ |
|-------------|------------------|
| 10500 | 1550 |
| 10519 | 1550 |
| 10529 | 1550 |
| 10539 | 1550 |
| 10549 | 1550 |
| 10559 | 1550 |

نتائج برنامج التمارين العامة تمرين رقم ٥ صفحة ٤٩٢

[illegible]

| | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | | 4 | | | | | | | | | | 5 | | | | | | | | | | 6 | | | | | | | | | | 7 | | | | | | | | | |
|-----|------------------|---|---|---|---|---|---|---|---|---|----------------------|---|---|---|---|---|---|---|---|---|-------------|---|---|---|---|---|---|---|---|---|----------------|---|---|---|---|---|---|---|---|---|---------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|--|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | BRIGHT STAR COMPANY | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | PAYROLL LIST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | DATE: 03/29/1959 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | EMPLOYEE NO. | | | | | | | | | | EMPLOYEE NAME | | | | | | | | | | BASE SALARY | | | | | | | | | | TRANSPORTATION | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | 2229 | | | | | | | | | | XXXXXXXXXXXXXXXXXXXX | | | | | | | | | | 2299.99 | | | | | | | | | | 229.99 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 28 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 29 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 30 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 32 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 33 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 34 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 35 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 36 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 37 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 38 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 39 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 40 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 41 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 42 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 43 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 44 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 45 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 46 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 47 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 48 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 49 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 50 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 51 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 52 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 53 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 55 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 56 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 57 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 59 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 60 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 61 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 62 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 63 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 64 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 65 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 66 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 67 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 68 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 69 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 70 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 71 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 72 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 73 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 74 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 75 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 76 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 77 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 78 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 79 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 80 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 81 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 82 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 83 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 84 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 85 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 86 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 87 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 88 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 89 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 90 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 91 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 92 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 93 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 94 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 95 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 96 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 97 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 98 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 99 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

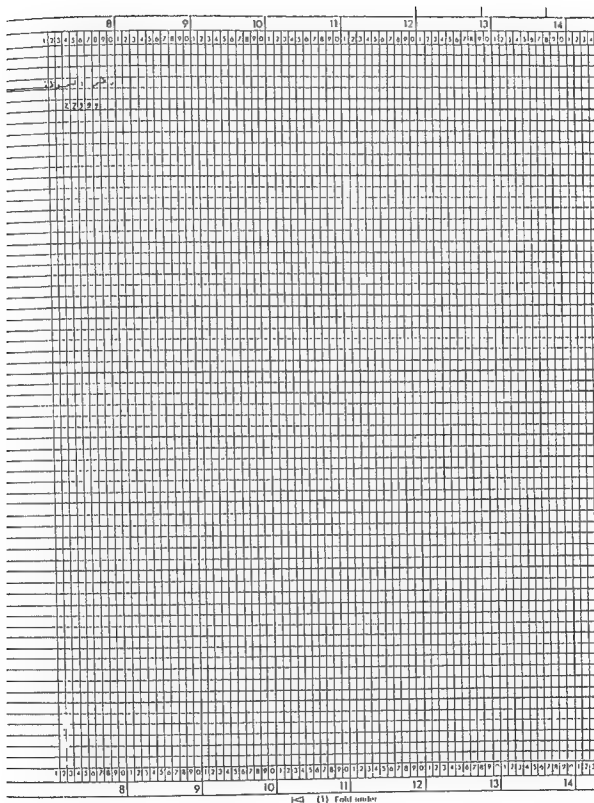
| | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | | 4 | | | | | | | | | | 5 | | | | | | | | | | 6 | | | | | | | | | | 7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2</ |

[illegible]

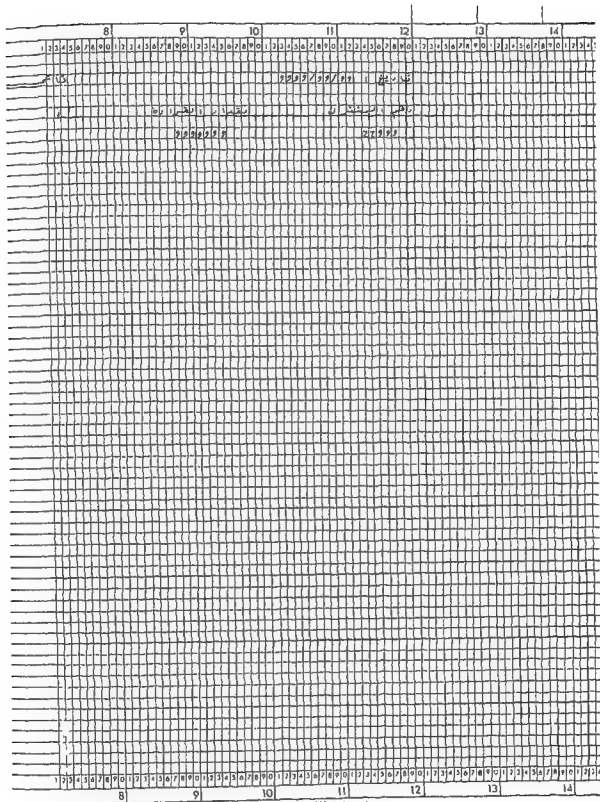
[illegible]

[illegible]

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 3 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 4 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 5 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 6 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 7 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 11 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 12 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 13 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 14 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 15 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 16 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 17 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 18 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 19 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 21 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 22 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 23 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 24 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 25 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 26 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 27 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 28 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 29 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 30 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 31 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 32 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 33 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 34 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 35 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 36 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 37 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 38 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 39 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 40 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 41 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 42 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 43 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 44 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 45 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 46 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 47 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 48 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 49 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 50 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 51 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 52 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 53 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 54 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 55 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 56 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 57 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 58 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 59 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 60 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 61 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 62 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 63 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 64 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 65 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 66 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 67 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 68 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 69 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 70 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 71 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 72 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 73 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 74 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 75 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 76 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 77 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 78 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 79 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 80 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 81 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 82 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 83 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 84 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 85 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 86 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 87 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 88 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 89 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 90 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 91 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 92 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 93 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 94 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 95 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 96 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 97 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 98 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 99 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 100 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |



| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 3 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 4 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 5 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |
| 6 | 36 | 37 | 38 | 39 | 40 | 41 | 42 |
| 7 | 43 | 44 | 45 | 46 | 47 | 48 | 49 |
| 8 | 50 | 51 | 52 | 53 | 54 | 55 | 56 |
| 9 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| 10 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 11 | 71 | 72 | 73 | 74 | 75 | 76 | 77 |
| 12 | 78 | 79 | 80 | 81 | 82 | 83 | 84 |
| 13 | 85 | 86 | 87 | 88 | 89 | 90 | 91 |
| 14 | 92 | 93 | 94 | 95 | 96 | 97 | 98 |
| 15 | 99 | 100 | 101 | 102 | 103 | 104 | 105 |
| 16 | 106 | 107 | 108 | 109 | 110 | 111 | 112 |
| 17 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| 18 | 120 | 121 | 122 | 123 | 124 | 125 | 126 |
| 19 | 127 | 128 | 129 | 130 | 131 | 132 | 133 |
| 20 | 134 | 135 | 136 | 137 | 138 | 139 | 140 |
| 21 | 141 | 142 | 143 | 144 | 145 | 146 | 147 |
| 22 | 148 | 149 | 150 | 151 | 152 | 153 | 154 |
| 23 | 155 | 156 | 157 | 158 | 159 | 160 | 161 |
| 24 | 162 | 163 | 164 | 165 | 166 | 167 | 168 |
| 25 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| 26 | 176 | 177 | 178 | 179 | 180 | 181 | 182 |
| 27 | 183 | 184 | 185 | 186 | 187 | 188 | 189 |
| 28 | 190 | 191 | 192 | 193 | 194 | 195 | 196 |
| 29 | 197 | 198 | 199 | 200 | 201 | 202 | 203 |
| 30 | 204 | 205 | 206 | 207 | 208 | 209 | 210 |
| 31 | 211 | 212 | 213 | 214 | 215 | 216 | 217 |
| 32 | 218 | 219 | 220 | 221 | 222 | 223 | 224 |
| 33 | 225 | 226 | 227 | 228 | 229 | 230 | 231 |
| 34 | 232 | 233 | 234 | 235 | 236 | 237 | 238 |
| 35 | 239 | 240 | 241 | 242 | 243 | 244 | 245 |
| 36 | 246 | 247 | 248 | 249 | 250 | 251 | 252 |
| 37 | 253 | 254 | 255 | 256 | 257 | 258 | 259 |
| 38 | 260 | 261 | 262 | 263 | 264 | 265 | 266 |
| 39 | 267 | 268 | 269 | 270 | 271 | 272 | 273 |
| 40 | 274 | 275 | 276 | 277 | 278 | 279 | 280 |
| 41 | 281 | 282 | 283 | 284 | 285 | 286 | 287 |
| 42 | 288 | 289 | 290 | 291 | 292 | 293 | 294 |
| 43 | 295 | 296 | 297 | 298 | 299 | 300 | 301 |
| 44 | 302 | 303 | 304 | 305 | 306 | 307 | 308 |
| 45 | 309 | 310 | 311 | 312 | 313 | 314 | 315 |
| 46 | 316 | 317 | 318 | 319 | 320 | 321 | 322 |
| 47 | 323 | 324 | 325 | 326 | 327 | 328 | 329 |
| 48 | 330 | 331 | 332 | 333 | 334 | 335 | 336 |
| 49 | 337 | 338 | 339 | 340 | 341 | 342 | 343 |
| 50 | 344 | 345 | 346 | 347 | 348 | 349 | 350 |
| 51 | 351 | 352 | 353 | 354 | 355 | 356 | 357 |
| 52 | 358 | 359 | 360 | 361 | 362 | 363 | 364 |
| 53 | 365 | 366 | 367 | 368 | 369 | 370 | 371 |
| 54 | 372 | 373 | 374 | 375 | 376 | 377 | 378 |
| 55 | 379 | 380 | 381 | 382 | 383 | 384 | 385 |
| 56 | 386 | 387 | 388 | 389 | 390 | 391 | 392 |
| 57 | 393 | 394 | 395 | 396 | 397 | 398 | 399 |
| 58 | 400 | 401 | 402 | 403 | 404 | 405 | 406 |
| 59 | 407 | 408 | 409 | 410 | 411 | 412 | 413 |
| 60 | 414 | 415 | 416 | 417 | 418 | 419 | 420 |
| 61 | 421 | 422 | 423 | 424 | 425 | 426 | 427 |
| 62 | 428 | 429 | 430 | 431 | 432 | 433 | 434 |
| 63 | 435 | 436 | 437 | 438 | 439 | 440 | 441 |
| 64 | 442 | 443 | 444 | 445 | 446 | 447 | 448 |
| 65 | 449 | 450 | 451 | 452 | 453 | 454 | 455 |
| 66 | 456 | 457 | 458 | 459 | 460 | 461 | 462 |
| 67 | 463 | 464 | 465 | 466 | 467 | 468 | 469 |
| 68 | 470 | 471 | 472 | 473 | 474 | 475 | 476 |
| 69 | 477 | 478 | 479 | 480 | 481 | 482 | 483 |
| 70 | 484 | 485 | 486 | 487 | 488 | 489 | 490 |
| 71 | 491 | 492 | 493 | 494 | 495 | 496 | 497 |
| 72 | 498 | 499 | 500 | 501 | 502 | 503 | 504 |
| 73 | 505 | 506 | 507 | 508 | 509 | 510 | 511 |
| 74 | 512 | 513 | 514 | 515 | 516 | 517 | 518 |
| 75 | 519 | 520 | 521 | 522 | 523 | 524 | 525 |
| 76 | 526 | 527 | 528 | 529 | 530 | 531 | 532 |
| 77 | 533 | 534 | 535 | 536 | 537 | 538 | 539 |
| 78 | 540 | 541 | 542 | 543 | 544 | 545 | 546 |
| 79 | 547 | 548 | 549 | 550 | 551 | 552 | 553 |
| 80 | 554 | 555 | 556 | 557 | 558 | 559 | 560 |
| 81 | 561 | 562 | 563 | 564 | 565 | 566 | 567 |
| 82 | 568 | 569 | 570 | 571 | 572 | 573 | 574 |
| 83 | 575 | 576 | 577 | 578 | 579 | 580 | 581 |
| 84 | 582 | 583 | 584 | 585 | 586 | 587 | 588 |
| 85 | 589 | 590 | 591 | 592 | 593 | 594 | 595 |
| 86 | 596 | 597 | 598 | 599 | 600 | 601 | 602 |
| 87 | 603 | 604 | 605 | 606 | 607 | 608 | 609 |
| 88 | 610 | 611 | 612 | 613 | 614 | 615 | 616 |
| 89 | 617 | 618 | 619 | 620 | 621 | 622 | 623 |
| 90 | 624 | 625 | 626 | 627 | 628 | 629 | 630 |
| 91 | 631 | 632 | 633 | 634 | 635 | 636 | 637 |
| 92 | 638 | 639 | 640 | 641 | 642 | 643 | 644 |
| 93 | 645 | 646 | 647 | 648 | 649 | 650 | 651 |
| 94 | 652 | 653 | 654 | 655 | 656 | 657 | 658 |
| 95 | 659 | 660 | 661 | 662 | 663 | 664 | 665 |
| 96 | 666 | 667 | 668 | 669 | 670 | 671 | 672 |
| 97 | 673 | 674 | 675 | 676 | 677 | 678 | 679 |
| 98 | 680 | 681 | 682 | 683 | 684 | 685 | 686 |
| 99 | 687 | 688 | 689 | 690 | 691 | 692 | 693 |
| 100 | 694 | 695 | 696 | 697 | 698 | 699 | 700 |



| 8 | | | | | | | | | | 9 | | | | | | | | | | 10 | | | | | | | | | | 11 | | | | | | | | | | 12 | | | | | | | | | | 13 | | | | | | | | | | 14 | | | | | | | | | | | | | | | | | | | |
|--------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |
| PAGE NO. 229 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| START DATE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| XXXXXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2595 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | | | | | | | | | | |

حقوق الطبع والنشر محفوظة لمعهد الإدارة العامة، ولا يجوز اقتباس
جزء من هذا الكتاب أو إعادة طبعه بآية صورة دون موافقة كتابية من إدارة البحوث
إلا في حالات الاقتباس القصيرة بفرض النقد والتحليل مع وجوب ذكر المصدر.

الإدارة العامة
مطبعة المطابع
١٤٠٥ هـ

جدول بالتصويبات الخاصة بكتاب البرجة التطبيقية بلغة الكوبول

| رقم الصفحة | رقم السطر | العبارة الخطأ | العبارة الصواب |
|------------|-----------|--|-------------------------------------|
| ٢٥ | ١٢ | - قسم البيانات | ٣ - قسم البيانات |
| ٩٥ | ٧ | أكبر من ٣٦ | أقل من ٣٦ |
| ١٠٤ | ٢١ | CONTAINS BO | CONTAINS 80 |
| ١٤٩ | ٧,٦ |] |] ,٠٠٠٠ |
| ١٥٧ | ١ | أجر العام | أجر العامل |
| ١٦٣ | ٥ | 217. | 217 |
| ١٧٥ | ٨ | 1234. | 12345. |
| ١٧٦ | ١ | مثال : | مثال : (١) |
| ١٧٦ | | يضاف لنهاية الصفحة | (١) قد تختلف الصيغة في بعض الأجهزة. |
| ١٨٦ | ٤ | OVERFLOW | OVERFLOW |
| ٢٠٨ | ٥ | - STORAG | - STORAGE |
| ٢٣٥ | ١٢ | رقم الطالب تصاعدياً | اسم الطالب تصاعدياً |
| ٢٤٦ | ٢ | - 4] | - 4] ,٠٠٠ |
| ٢٧٨ | ١٠ | READ MSTR | READ TRNS |
| ٤٠٥ | ٨ | وسائل تخزين | وسائل تخزين |
| ٤١٨ | ١٠ | COMMAND > INPUT | COMMAND INPUT > |
| ٤٦٢ | | الشكل أسفل الصفحة | عمليات الحذف |
| ٤٦٢ | | الشكل أسفل الصفحة | عمليات تعديل |
| ٥١٣ | ٦ | تخذف الأسطر اعتباراً من السطر ٦ لنهاية الصفحة لتكرارها | |
| ٥٢٤ | ١ | التقطعة (الفاصلة العشرية) | . التقطعة (الفاصلة العشرية) |
| ٥٢٤ | ٢ | الحظ المائل | / الحظ المائل |

Bibliotheca Alexandrina



0242085

السعر ٣٤ ريالاً



١٤١٠ هـ